

(19) 日本国特許庁 (JP)

## (12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2004-304767

(P2004-304767A)

(43) 公開日 平成16年10月28日 (2004. 10. 28)

(51) Int. Cl. <sup>7</sup>

HO4N 5/92  
G11B 20/10  
G11B 20/12  
G11B 27/00  
G11B 27/10

F I

HO4N 5/92  
G11B 20/10  
G11B 20/12  
G11B 27/00  
G11B 27/10

3 2 1 Z

D

A

テーマコード (参考)

5C052  
5C053  
5D044  
5D077  
5D110

審査請求 未請求 請求項の数 88 O L (全 108 頁) 最終頁に続く

(21) 出願番号 特願2004-21886 (P2004-21886)  
(22) 出願日 平成16年1月29日 (2004. 1. 29)  
(31) 優先権主張番号 特願2003-22551 (P2003-22551)  
(32) 優先日 平成15年1月30日 (2003. 1. 30)  
(33) 優先権主張国 日本国 (JP)  
(31) 優先権主張番号 特願2003-74441 (P2003-74441)  
(32) 優先日 平成15年3月18日 (2003. 3. 18)  
(33) 優先権主張国 日本国 (JP)

(71) 出願人 000002185  
ソニー株式会社  
東京都品川区北品川6丁目7番35号  
(74) 代理人 100082762  
弁理士 杉浦 正知  
(74) 代理人 100120640  
弁理士 森 幸一  
(72) 発明者 加藤 元樹  
東京都品川区北品川6丁目7番35号 ソ  
ニー株式会社内  
(72) 発明者 浜田 俊也  
東京都品川区北品川6丁目7番35号 ソ  
ニー株式会社内  
Fターム (参考) 5C052 AA02

最終頁に続く

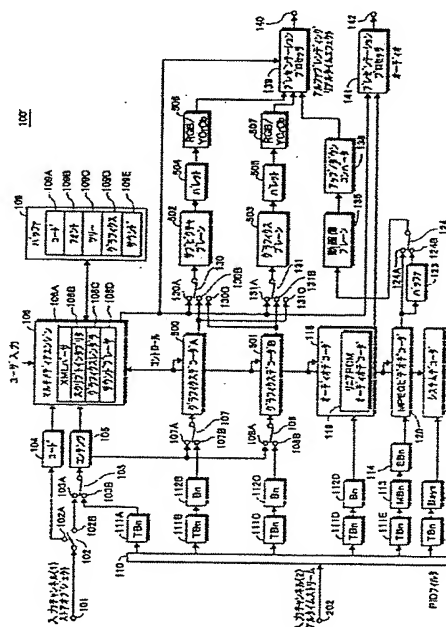
(54) 【発明の名称】 再生装置、再生方法、再生プログラムおよび記録媒体

## (57) 【要約】

【課題】 コンテンツ記録済み大容量ディスク状記録媒体において、自由度の高いユーザインターフェイスを実現する。

【解決手段】 ボタン画像データおよび対応する効果音の音声データと、音声データや画像データの制御命令とをストリームに多重化してディスクに記録する。ディスクから再生したシナリオをバッファ104に格納し、画像データおよび音声データをバッファ105に夫々溜め込む。シナリオに従いバッファ105から画像データを読み出してグラフィクスプレーン503に展開する。それと共に、対応する音声データがバッファ105から読み出されサウンドプレーヤ106Dにより再生される。ボタン状態の変化に応じたボタン画像の変更に伴い効果音が再生され、より効果的なユーザインターフェイスが実現される。

【選択図】 図69



## 【特許請求の範囲】

## 【請求項 1】

円盤状記録媒体に記録されたコンテンツデータを再生する再生装置において、  
記録媒体から再生された、少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画データおよび字幕データを含むリアルタイムストリームとが入力される入力手段と、  
上記入力手段により入力された上記プログラムコードを格納するコード格納手段と、  
上記入力手段により入力された上記画像データを格納する画像データ格納手段と、  
上記入力手段により入力された上記動画データをデコードした復号動画データと、上記入力手段により入力された上記字幕データをデコードした復号字幕データとを合成する第 10  
1の合成手段と、

上記コード格納手段に格納された上記プログラムコードに基づき、上記画像データ格納手段に格納された上記画像データをデコードした復号画像データと、上記第 1の合成手段により合成された動画字幕合成データとを合成する第 2の合成手段と  
を有することを特徴とする再生装置。

## 【請求項 2】

請求項 1 に記載の再生装置において、  
上記画像データの上記デコードは、上記プログラムコードに基づきユーザからの入力に応じて行われることを特徴とする再生装置。

## 【請求項 3】

請求項 1 に記載の再生装置において、  
上記第 2の合成手段による上記合成は、上記プログラムコードに基づきユーザからの入力に応じて行われることを特徴とする再生装置。

## 【請求項 4】

請求項 1 に記載の再生装置において、  
上記プログラムコードは、HTMLまたは XHTML に基づき記述されていることを特徴とする再生装置。

## 【請求項 5】

請求項 1 に記載の再生装置において、  
上記プログラムコードは、ECMAScript で記述されていることを特徴とする再生 30  
装置。

## 【請求項 6】

請求項 1 に記載の再生装置において、  
上記復号動画データの解像度を変換する解像度変換手段をさらに有し、上記第 1の合成手段は、上記解像度変換手段の出力と上記復号字幕データとを合成するようにしたことを特徴とする再生装置。

## 【請求項 7】

請求項 1 に記載の再生装置において、  
上記第 1の合成手段による上記合成の度合いは、上記復号字幕データに応じて制御されることを特徴とする再生装置。

## 【請求項 8】

請求項 1 に記載の再生装置において、  
上記第 2の合成手段による上記合成の度合いは、上記復号画像データに応じて制御されることを特徴とする再生装置。

## 【請求項 9】

請求項 1 に記載の再生装置において、  
上記復号字幕データの色情報を上記画像字幕合成データに用いる色として YCbCr 形式の色情報として出力させる第 1の色情報変換手段をさらに有し、上記第 1の合成手段は、上記第 1の色情報変換手段で色情報が変換された上記復号字幕データと、上記復号動画データとを合成するようにしたことを特徴とする再生装置。

20

40

50

## 【請求項 10】

請求項 1 に記載の再生装置において、

上記復号画像データの色情報を RGB 形式から YCbCr 形式に変換する第 2 の色情報変換手段をさらに有し、上記第 2 の合成手段は、上記第 2 の色情報変換手段で色情報が変換された上記復号画像データと、上記動画字幕合成データとを合成するようにしたことを特徴とする再生装置。

## 【請求項 11】

請求項 1 に記載の再生装置において、

上記リアルタイムストリームは、MPEG2 TS のトランスポートパケットに格納されて上記記録媒体から再生されることを特徴とする再生装置。

10

## 【請求項 12】

請求項 1 に記載の再生装置において、

上記リアルタイムストリームに上記画像データがさらに含まれることを特徴とする再生装置。

## 【請求項 13】

請求項 1 に記載の再生装置において、

上記入力手段により入力された 1 または複数の上記字幕データを格納する字幕データ格納手段をさらに有し、

上記第 1 の合成手段は、上記字幕データ格納手段に格納された複数の上記字幕データをデコードした複数の上記復号字幕データと、上記復号動画データとを合成できるようにしたことを特徴とする再生装置。

20

## 【請求項 14】

請求項 1 に記載の再生装置において、

上記画像データ格納手段は、1 または複数の上記画像データを格納するようにされ、

上記第 2 の合成手段は、上記画像データ格納手段に格納された複数の上記画像データを復号した複数の上記復号画像データと、上記動画字幕合成データとを合成できるようにしたことを特徴とする再生装置。

## 【請求項 15】

請求項 1 に記載の再生装置において、

上記画像データはボタンを表示するボタン画像データであって、上記ボタンの 3 種類の状態にそれぞれ対応する第 1、第 2 および第 3 のボタン画像データをユーザの上記操作に従い差し替えて表示するようにしたことを特徴とする再生装置。

30

## 【請求項 16】

請求項 1 に記載の再生装置において、

上記画像データおよび上記字幕データは、上記動画データに対して同期的に表示制御され、共通のデータ構造に格納されることを特徴とする再生装置。

## 【請求項 17】

請求項 16 に記載の再生装置において、

上記データ構造は、少なくとも、表示制御命令と、該表示制御命令により表示制御される 1 または複数の上記画像データまたは上記字幕データとを有することを特徴とする再生装置。

40

## 【請求項 18】

請求項 17 に記載の再生装置において、

上記データ構造は、同時に実行される複数の上記表示制御命令をひとまとめにして表示制御命令群として格納し、それぞれ異なる時刻に実行される複数の上記表示制御命令群を格納できるようにしたことを特徴とする再生装置。

## 【請求項 19】

請求項 16 に記載の再生装置において、

上記データ構造に格納される上記画像データまたは上記字幕データは、上記画像データをデコードする際の時間管理情報または上記字幕データをデコードする際の時間管理情報

50

に基づき有効期間が開始され、該データ構造に定義される有効期間終了時刻で有効期間が終了されることを特徴とする再生装置。

【請求項 20】

請求項 16 に記載の再生装置において、

上記画像データはボタンを表示するボタン画像データであって、上記ボタンの 3 種類の状態にそれぞれ対応する第 1、第 2 および第 3 のボタン画像データを同一の上記データ構造に格納するようにしたことを特徴とする再生装置。

【請求項 21】

請求項 17 に記載の再生装置において、

上記表示制御命令は、音声データの再生を制御する音声データ再生制御命令をさらに含めるようにしたことを特徴とする再生装置。 10

【請求項 22】

請求項 21 に記載の再生装置において、

上記音声データ再生制御命令に基づき上記字幕データの表示に対して同期的に音声データの再生を制御するようにしたことを特徴とする再生装置。

【請求項 23】

請求項 21 に記載の再生装置において、

上記画像データはボタンを表示するボタン画像データであって、上記音声データ再生制御命令に基づき上記ボタンの 3 種類の状態にそれぞれ対応する第 1、第 2 および第 3 のボタン画像データの表示に応じて上記音声データを再生するようにしたことを特徴とする再生装置。 20

【請求項 24】

請求項 21 に記載の再生装置において、

上記音声データ再生制御命令により、上記画像データまたは上記字幕データに対して上記音声データを割り当てることができるようにしたことを特徴とする再生装置。

【請求項 25】

請求項 1 に記載の再生装置において、

上記入力手段に対して、効果音を再生する音声データがさらに入力されることを特徴とする再生装置。

【請求項 26】

請求項 25 に記載の再生装置において、

上記音声データは、上記リアルタイムストリームに多重化されて入力されることを特徴とする再生装置。 30

【請求項 27】

請求項 26 に記載の再生装置において、

上記リアルタイムストリームは画像データがさらに多重化され、該リアルタイムストリームに含まれ多重化される画像データおよび上記音声データは、1 のデータ構造にまとめて格納されて入力されることを特徴とする記録媒体。

【請求項 28】

請求項 27 に記載の再生装置において、

上記データ構造には、上記リアルタイムストリームに多重化される画像データの表示制御および上記音声データの再生制御を行う制御命令がさらに格納されていることを特徴とする再生装置。 40

【請求項 29】

請求項 27 に記載の再生装置において、

上記データ構造には、上記リアルタイムストリームに多重化される画像データと対応する上記音声データが一组とされて格納されていることを特徴とする再生装置。

【請求項 30】

請求項 29 に記載の再生装置において、

上記リアルタイムストリームに多重化される画像データの表示に応じて該画像データと 50



対応する上記音声データが再生されるようにしたことを特徴とする再生装置。

【請求項 3 1】

請求項 3 0 に記載の再生装置において、

上記リアルタイムストリームに多重化される画像データは、ボタンを表示するボタン画像データであって、上記データ構造には、上記ボタンの状態に対応した上記ボタン画像データと上記音声データとが一组とされて格納されていることを特徴とする再生装置。

【請求項 3 2】

請求項 3 1 に記載の再生装置において、

上記ボタンは複数の状態を有し、該複数の状態に複数のボタン画像データがそれぞれ対応付けられると共に、該複数のボタン画像データの少なくとも 1 つに音声データが対応付けられて一組とされて、該複数のボタン画像データと該音声データとが同一の上記データ構造に格納されていることを特徴とする再生装置。

【請求項 3 3】

請求項 3 2 に記載の再生装置において、

上記ボタンの上記状態の変化に伴う上記ボタン画像データの表示に応じて該ボタン画像データと一組をなす上記音声データを再生するようにしたことを特徴とする再生装置。

【請求項 3 4】

請求項 3 3 に記載の再生装置において、

上記データ構造は、同時に実行される複数の上記制御命令をひとまとめにして制御命令群として格納し、それぞれ異なる時刻に実行される複数の上記制御命令群を格納できるようにされたことを特徴とする再生装置。

【請求項 3 5】

請求項 2 5 に記載の再生装置において、

上記音声データは、ファイルに格納されて上記非リアルタイムストリームとして上記入力手段から入力されることを特徴とする再生装置。

【請求項 3 6】

請求項 3 5 に記載の再生装置において、

上記リアルタイムストリームに多重化される画像データと、上記音声データを示す情報とが、1 のデータ構造にまとめて格納されて入力されるようにしたことを特徴とする再生装置。

【請求項 3 7】

請求項 3 6 に記載の再生装置において、

上記データ構造は、上記リアルタイムストリームに多重化される画像データと、該画像データと対応する上記音声データを示す情報とが一组とされて格納され、該画像データの表示に応じて該画像データに対応する上記音声データが再生されるようにしたことを特徴とする再生装置。

【請求項 3 8】

請求項 3 7 に記載の再生装置において、

上記リアルタイムストリームに多重化される画像データは、ボタンを表示するボタン画像データであって、上記データ構造には、上記ボタンの状態に対応した上記ボタン画像データと、該ボタン画像データと対応する上記音声データを示す情報とが一组とされて格納されていることを特徴とする再生装置。

【請求項 3 9】

請求項 3 8 に記載の再生装置において、

上記ボタンは複数の状態を有し、該複数の状態に複数のボタン画像データがそれぞれ対応付けられると共に、該複数のボタン画像データの少なくとも 1 つに音声データが対応付けられて一組とされ、該複数のボタン画像データと該音声データを示す情報とが同一の上記データ構造に格納されていることを特徴とする再生装置。

【請求項 4 0】

請求項 3 9 に記載の再生装置において、

上記データ構造は、上記ボタン画像データの表示に応じて該ボタン画像データと対になる上記音声データを再生するようにした制御命令がさらに格納され、上記制御命令に基づき上記音声データを再生するようにしたことを特徴とする再生装置。

【請求項 4 1】

請求項 4 0 に記載の再生装置において、

上記データ構造は、同時に実行される複数の上記制御命令をひとまとめにして制御命令群として格納し、それぞれ異なる時刻に実行される複数の上記制御命令群を格納できるようにされていることを特徴とする再生装置。

【請求項 4 2】

請求項 3 5 に記載の再生装置において、

上記ファイルは、複数の上記音声データがまとめて格納されることを特徴とする再生装置。

10

【請求項 4 3】

請求項 4 2 に記載の再生装置において、

上記ファイルは、さらに、該ファイルに格納される上記複数の音声データそれぞれの該ファイル中での位置情報が、該音声データの識別情報に対応付けられて格納されることを特徴とする再生装置。

【請求項 4 4】

請求項 4 3 に記載の再生装置において、

上記音声データが上記識別情報により上記画像データと関連付けられ、上記画像データの表示に伴い、該画像データに関連付けられた上記音声データが上記識別情報に基づき上記ファイルから読み出されるようにしたことを特徴とする再生装置。

20

【請求項 4 5】

請求項 4 2 に記載の再生装置において、

上記ファイルは、上記プログラムコードの実行に先立ち予め上記入力手段から入力されることを特徴とする再生装置。

【請求項 4 6】

請求項 4 2 に記載の再生装置において、

複数の上記ファイルが上記入力手段から入力されることを特徴とする再生装置。

【請求項 4 7】

請求項 4 6 に記載の再生装置において、

上記複数のファイルのそれぞれは、上記プログラムコードの再生単位で該プログラムコードから参照され、該参照毎に上記入力手段から入力されることを特徴とする再生装置。

30

【請求項 4 8】

請求項 1 に記載の再生装置において、

上記画像データをデコードする際のレートの上限を規定するようにしたことを特徴とする再生装置。

【請求項 4 9】

請求項 1 に記載の再生装置において、

上記復号画像データをブレンバッファに転送する際の転送レートの上限を規定するようにしたことを特徴とする再生装置。

40

【請求項 5 0】

円盤状記録媒体に記録されたコンテンツデータを再生する再生方法において、

記録媒体から再生された、少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画データおよび字幕データを含むリアルタイムストリームとを入力する入力のステップと、

上記入力のステップにより入力された上記プログラムコードをコード格納手段に格納するステップと、

上記入力のステップにより入力された上記画像データを画像データ格納手段に格納するステップと、

50

上記入力の手続きにより入力された上記動画データをデコードした復号動画データと、上記入力の手続きにより入力された上記字幕データをデコードした復号字幕データとを合成する第1の合成の手続きと、

上記コード格納手段に格納された上記プログラムコードに基づき、上記画像データ格納手段に格納された上記画像データをデコードした復号画像データと、上記第1の合成の手続きにより合成された動画字幕合成データとを合成する第2の合成の手続きとを有することを特徴とする再生方法。

【請求項51】

円盤状記録媒体に記録されたコンテンツデータを再生する再生方法をコンピュータ装置に実行させる再生プログラムにおいて、

記録媒体から再生された、少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画データおよび字幕データを含むリアルタイムストリームとを入力する入力の手続きと、

上記入力の手続きにより入力された上記プログラムコードをコード格納手段に格納する手続きと、

上記入力の手続きにより入力された上記画像データを画像データ格納手段に格納する手続きと、

上記入力の手続きにより入力された上記動画データをデコードした復号動画データと、上記入力の手続きにより入力された上記字幕データをデコードした復号字幕データとを合成する第1の合成の手続きと、

上記コード格納手段に格納された上記プログラムコードに基づき、上記画像データ格納手段に格納された上記画像データをデコードした復号画像データと、上記第1の合成の手続きにより合成された動画字幕合成データとを合成する第2の合成の手続きとを有する再生方法をコンピュータ装置に実行させることを特徴とする再生プログラム。

【請求項52】

円盤状記録媒体に記録されたコンテンツデータを再生する再生方法をコンピュータ装置に実行させる再生プログラムが記録された記録媒体において、

記録媒体から再生された、少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画データおよび字幕データを含むリアルタイムストリームとを入力する入力の手続きと、

上記入力の手続きにより入力された上記プログラムコードをコード格納手段に格納する手続きと、

上記入力の手続きにより入力された上記画像データを画像データ格納手段に格納する手続きと、

上記入力の手続きにより入力された上記動画データをデコードした復号動画データと、上記入力の手続きにより入力された上記字幕データをデコードした復号字幕データとを合成する第1の合成の手続きと、

上記コード格納手段に格納された上記プログラムコードに基づき、上記画像データ格納手段に格納された上記画像データをデコードした復号画像データと、上記第1の合成の手続きにより合成された動画字幕合成データとを合成する第2の合成の手続きとを有する再生方法をコンピュータ装置に実行させる再生プログラムが記録されたことを特徴とする記録媒体。

【請求項53】

コンテンツデータが記録された円盤状の形状を有する記録媒体において、

少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画データおよび字幕データを含むリアルタイムストリームとが記録され、

上記プログラムコードに基づき、再生後に画像データ格納手段に格納された上記画像データをデコードした復号画像データと、再生された上記動画データをデコードした復号動画データおよび再生された上記字幕データをデコードした復号字幕データが合成された動

画字幕合成データとが合成されることを特徴とする記録媒体。

【請求項 5 4】

請求項 5 3 に記載の記録媒体において、

上記画像データはボタンを表示するボタン画像データであって、上記ボタンの 3 種類の状態にそれぞれ対応する第 1、第 2 および第 3 のボタン画像データをユーザの上記操作に従い差し替えて表示するようにしたことを特徴とする記録媒体。

【請求項 5 5】

請求項 5 3 に記載の記録媒体において、

上記画像データおよび上記字幕データは、上記動画データに対して同期的に表示制御され、共通のデータ構造に格納されて記録されることを特徴とする記録媒体。

10

【請求項 5 6】

請求項 5 5 に記載の記録媒体において、

上記データ構造は、少なくとも、表示制御命令と、該表示制御命令により表示制御される 1 または複数の上記画像データまたは上記字幕データとを有することを特徴とする記録媒体。

【請求項 5 7】

請求項 5 6 に記載の記録媒体において、

上記データ構造は、同時に実行される複数の上記表示制御命令をひとまとめにして表示制御命令群として格納し、それぞれ異なる時刻に実行される複数の上記表示制御命令群を格納できるようにして記録されることを特徴とする記録媒体。

20

【請求項 5 8】

請求項 5 5 に記載の記録媒体において、

上記画像データをデコードする際の時間管理情報または上記字幕データをデコードする際の時間管理情報に基づき有効期間が開始され、上記データ構造に定義される有効期間終了時刻で有効期間が終了するようにした上記画像データまたは上記字幕データが上記データ構造に格納されて記録されることを特徴とする記録媒体。

【請求項 5 9】

請求項 5 5 に記載の記録媒体において、

上記画像データはボタンを表示するボタン画像データであって、上記ボタンの 3 種類の状態にそれぞれ対応する第 1、第 2 および第 3 のボタン画像データを同一の上記データ構造に格納して記録することを特徴とする記録媒体。

30

【請求項 6 0】

請求項 5 6 に記載の記録媒体において、

上記表示制御命令は、音声データの再生を制御する音声データ再生制御命令をさらに含めて上記データ構造に格納され記録されることを特徴とする記録媒体。

【請求項 6 1】

請求項 6 0 に記載の記録媒体において、

上記音声データ再生制御命令に基づき上記字幕データの表示に対して同期的に音声データの再生を制御するようにした上記表示制御命令が上記データ構造に格納されて記録されることを特徴とする記録媒体。

40

【請求項 6 2】

請求項 6 0 に記載の記録媒体において、

上記画像データはボタンを表示するボタン画像データであって、上記音声データ再生制御命令に基づき上記ボタンの 3 種類の状態にそれぞれ対応する第 1、第 2 および第 3 のボタン画像データの表示に応じて上記音声データを再生するようにした上記表示制御命令が上記データ構造に格納されて記録されることを特徴とする記録媒体。

【請求項 6 3】

請求項 6 0 に記載の記録媒体において、

上記音声データ再生制御命令により上記画像データまたは上記字幕データに対して上記音声データを割り当てることができるようにした上記表示制御命令が上記データ構造に格

50

納されて記録されることを特徴とする記録媒体。

【請求項 64】

請求項 53 に記載の記録媒体において、

上記画像データがさらに上記リアルタイムストリームに多重化されて記録されることを特徴とする記録媒体。

【請求項 65】

請求項 53 に記載の記録媒体において、

効果音を再生する音声データがさらに記録されることを特徴とする記録媒体。

【請求項 66】

請求項 65 に記載の記録媒体において、

上記音声データは、上記リアルタイムストリームに多重化されて記録されることを特徴とする記録媒体。

10

【請求項 67】

請求項 66 に記載の記録媒体において、

上記リアルタイムストリームは画像データがさらに多重化され、該リアルタイムストリームに含まれる画像データおよび上記音声データを、1 のデータ構造にまとめて格納して記録するようにしたことを特徴とする記録媒体。

【請求項 68】

請求項 67 に記載の記録媒体において、

上記データ構造には、上記リアルタイムストリームに多重化される画像データの表示制御および上記音声データの再生制御を行う制御命令がさらに格納されることを特徴とする記録媒体。

20

【請求項 69】

請求項 67 に記載の記録媒体において、

上記データ構造には、上記リアルタイムストリームに多重化される画像データに対応する上記音声データが一組とされて格納されることを特徴とする記録媒体。

【請求項 70】

請求項 69 に記載の記録媒体において、

上記リアルタイムストリームに多重化される画像データは、ボタンを表示するボタン画像データであって、上記データ構造には、上記ボタンの状態に対応した上記ボタン画像データと上記音声データとが一組とされて格納されることを特徴とする記録媒体。

30

【請求項 71】

請求項 70 に記載の記録媒体において、

上記ボタンは複数の状態を有し、該複数の状態に複数のボタン画像データがそれぞれ対応付けられると共に、該複数のボタン画像データの少なくとも1 つに音声データが対応付けられ一組とされて、該複数のボタン画像データと該音声データとを同一の上記データ構造に格納することを特徴とする記録媒体。

【請求項 72】

請求項 71 に記載の記録媒体において、

上記ボタンの上記状態の変化に伴う上記ボタン画像データの表示に応じて該ボタン画像データと一組をなす上記音声データを再生するようにされていることを特徴とする記録媒体。

40

【請求項 73】

請求項 72 に記載の記録媒体において、

上記データ構造は、同時に実行される複数の上記制御命令をひとまとめにして制御命令群として格納し、それぞれ異なる時刻に実行される複数の上記制御命令群を格納できるようにして記録されることを特徴とする記録媒体。

【請求項 74】

請求項 65 に記載の記録媒体において、

上記音声データは、ファイルに格納されて記録されることを特徴とする記録媒体。

50

**【請求項 7 5】**

請求項 7 4 に記載の記録媒体において、

上記リアルタイムストリームに多重化される画像データと、上記音声データを示す情報とを、1 のデータ構造にまとめて格納して記録するようにしたことを特徴とする記録媒体

。

**【請求項 7 6】**

請求項 7 5 に記載の記録媒体において、

上記データ構造には、上記リアルタイムストリームに多重化される画像データと、該画像データと対応する上記音声データを示す情報とが一组とされて格納されることを特徴とする記録媒体。

10

**【請求項 7 7】**

請求項 7 6 に記載の記録媒体において、

上記リアルタイムストリームに多重化される画像データは、ボタンを表示するボタン画像データであって、上記データ構造には、上記ボタンの状態に対応した上記ボタン画像データと、該ボタン画像データと対応する上記音声データを示す情報とが一组とされて格納されることを特徴とする記録媒体。

**【請求項 7 8】**

請求項 7 7 に記載の記録媒体において、

上記ボタンは複数の状態を有し、該複数の状態に複数のボタン画像データがそれぞれ対応付けられると共に、該複数のボタン画像データの少なくとも 1 つに音声データが対応付けられ一組とされて、該複数のボタン画像データと該音声データを示す情報とが同一の上記データ構造に格納されることを特徴とする記録媒体。

20

**【請求項 7 9】**

請求項 7 8 に記載の記録媒体において、

上記ボタン画像データの表示に応じて該ボタン画像データと対になる上記音声データを再生するようにした制御命令が上記データ構造にさらに格納されることを特徴とする記録媒体。

**【請求項 8 0】**

請求項 7 9 に記載の記録媒体において、

上記データ構造は、同時に実行される複数の上記制御命令をひとまとめにして制御命令群として格納し、それぞれ異なる時刻に実行される複数の上記制御命令群を格納できるようにして記録されることを特徴とする記録媒体。

30

**【請求項 8 1】**

請求項 6 5 に記載の記録媒体において、

上記ファイルは、複数の上記音声データがまとめて格納されることを特徴とする記録媒体。

**【請求項 8 2】**

請求項 8 1 に記載の記録媒体において、

上記ファイルは、さらに、該ファイルに格納される上記複数の音声データそれぞれの該ファイル中での位置情報が、該音声データの識別情報に対応付けられて格納されることを特徴とする記録媒体。

40

**【請求項 8 3】**

請求項 8 2 に記載の記録媒体において、

上記音声データが上記識別情報により上記画像データと関連付けられ、上記画像データの表示に伴い、該画像データに関連付けられた上記音声データが上記識別情報に基づき上記ファイルから読み出されるようにしたことを特徴とする記録媒体。

**【請求項 8 4】**

請求項 8 1 に記載の記録媒体において、

上記ファイルは、上記プログラムコードの実行に先立ち予め読み出されるようにしたことを特徴とする記録媒体。

50

## 【請求項 85】

請求項 81 に記載の記録媒体において、  
複数の上記ファイルが記録されることを特徴とする記録媒体。

## 【請求項 86】

請求項 84 に記載の記録媒体において、  
上記複数のファイルのそれぞれは、上記プログラムコードの再生単位で該プログラムコードから参照され、該参照毎に読み出されるようにしたことを特徴とする記録媒体。

## 【請求項 87】

請求項 53 に記載の記録媒体において、  
予め規定された、上記画像データをデコードする際のレートの下限を満足するようにして 10  
て上記画像データが記録されることを特徴とする記録媒体。

## 【請求項 88】

請求項 53 に記載の記録媒体において、  
予め規定された、上記復号画像データをプレーンバッファに転送する際の転送レートの下限を満足するようにして上記画像データが記録されることを特徴とする記録媒体。

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

この発明は、ブルーレイディスク(Blu-ray Disc)といった大容量の記録媒体に記録されたプログラムに対するユーザによるインタラクティブな操作を可能とする再生装置、再生 20  
方法、再生プログラムおよび記録媒体に関する。

## 【背景技術】

## 【0002】

近年、記録可能で記録再生装置から取り外し可能なディスク型記録媒体の規格として、Blu-ray Disc (ブルーレイディスク) 規格が提案されている。Blu-ray Disc 規格では、記録媒体として直径 12 cm、カバー層 0.1 mm のディスクを用い、光学系として波長 405 nm の青紫色レーザ、開口数 0.85 の対物レンズを用いて、最大で 27 GB (ギガバイト) の記録容量を実現している。これにより、日本の BS デジタルハイビジョン放送を、画質を劣化させることなく 2 時間以上記録することが可能である。 30

## 【0003】

この記録可能光ディスクに記録する AV (Audio/Video) 信号のソース (供給源) としては、従来からの、例えばアナログテレビジョン放送によるアナログ信号によるものと、例えば BS デジタル放送をはじめとするデジタルテレビジョン放送によるデジタル信号によるものとが想定されている。Blu-ray Disc 規格では、これらの放送による AV 信号を記録する方法を定めた規格は、既に作られている。

## 【0004】

一方で、現状の Blu-ray Disc のの派生規格として、映画や音楽などが予め記録された、再生専用の記録媒体を開発する動きが進んでいる。映画や音楽を記録するためのディスク状記録媒体としては、既に DVD (Digital Versatile Disc) が広く普及しているが、この Blu-ray Disc の企画に基づいた再生専用光ディスクは、Blu-ray Disc の大容量および高速な転送速度などを活かし、ハイビジョン映像を高画質なままで 2 時間以上収録できる点が、既存の DVD とは大きく異なり、優位である。 40

## 【0005】

一方で、現状の Blu-ray Disc の規格では、ディスクに記録されている映像コンテンツの一覧を画面表示する方法や、その一覧表上にカーソルを表示させ、再生したい映像コンテンツをユーザに選択させるなどといったユーザインターフェイスに関する機能が定められていない。これらの機能は、Blu-ray Disc に対する記録再生を行う記録再生装置本体によって実現されている。そのため、同一の記録媒体を再生した場合でも、再生に用いた記録再生装置によってコンテンツ一覧画面のレイアウトが異なって 50

しまい、ユーザインタフェースにも差が生じ、必ずしもユーザにとって使い易いものではない。再生専用ディスクとしては、再生機器によらず、ディスク（コンテンツ）制作者が意図した通りのメニュー画面などが表示され、意図通りのユーザインタフェースが実現される必要がある。

#### 【0006】

また、映像コンテンツの再生中に選択画面が表示され、ユーザの選択によってストーリーが分岐していくマルチストーリーの機能は、一般にインタラクティブ機能とも呼ばれる。このインタラクティブ機能を実現するためには、ディスク制作者が再生順序や分岐を定めたシナリオを作り、そのシナリオをプログラム言語、スクリプト言語等を使って記述し、ディスクに記録しておく必要がある。再生装置側では、そのプログラムを読み込み、実行することで、制作者の意図に従った映像コンテンツの再生や、分岐のための選択画面提示を実現することになる。

#### 【0007】

このように、現状のBlu-ray Disc規格(Blu-ray Disc Rewritable Format Ver1.0)では、この制作者の意図通りのユーザインタフェースを実現するための、メニュー画面や分岐選択画面の構成方法、ユーザ入力に対する処理を記述する方法が定められていない。そのため、現状では、Blu-ray Discを用いて、制作者が意図したシナリオ通りの再生を、再生装置の製造メーカーや機種に左右されることなく互換性を持たせた形で実現することが難しい。

#### 【0008】

また、映画を収録した再生専用ディスクにおいては、字幕を表示する仕組みが不可欠である。しかしながら、この字幕表示についても、現状のBlu-ray Disc規格では、定められていない。

#### 【0009】

一方、従来から、例えばDVD(Digital Versatile Disc)の規格においては、上述のようなインタラクティブな機能が既に実現されていた。例えば、DVDビデオにより動画を再生中に、リモートコントロールコマンドなどを用いてメニュー画面を呼び出し、例えばメニュー画面上に配置されたボタンを選択するなどして、再生場面を変更するなどの処理が可能であった。また、字幕を表示する仕組みも規定されていた。字幕表示については、例えば、予め用意されている日本語字幕と英語字幕とを切り換えて表示させることができた。

#### 【0010】

DVDの場合、メニュー画面を固定的なサブピクチャデータにより構成し、メニュー画面が呼び出された際に、動画データにこのサブピクチャデータを合成して表示する。特許文献1に、このように動画データにサブピクチャデータを合成して記録可能なDVDに記録する構成が記載されている。

【特許文献1】特開平10-308924号公報

#### 【0011】

ここで、従来技術によるメニュー画面の一例の表示について、概略的に説明する。例えば、映画が収録されたDVDを再生装置で再生する際に、映画本編が再生されるのに先立って、メニュー画面が表示される。一般に、メニュー画面には、複数のボタンが配置される。各ボタンには、それぞれ特定の動作が割り当てられており、ボタンを選択し実行を指示すると、選択されたボタンに割り当てられた動作が実行される。例えば、「映画本編」と表示されたボタンを選択し実行を指示することで、当該ボタンに割り当てられた動作が実行され、当該DVDに収録された映画の本編が再生されることになる。

#### 【0012】

ユーザは、例えば、リモートコントロールコマンド（以下、リモコンと略称する）に設けられた、上下左右の方向をそれぞれ指示するようにされたキー（キー（方向キー）などを操作して、メニュー画面に表示されたボタンの一つを選択状態とし、そこで決定キーを操作することで、選択されたボタンに割り当てられた動作を実行状態とする。また、ボタ



ンには、通常（非選択）状態、選択状態及び実行状態という３種類の状態が存在し、ユーザが区別し易いように、それぞれ画像や色などが異ならされている。一般に、選択状態および実行状態のボタンは、同時には一つしか存在しない。

#### 【0013】

例えばDVDビデオの場合、ボタンの表示は、それぞれサブピクチャ、ハイライトと呼ばれる２種類のデータにより実現される。図86は、従来技術による一例のDVDメニュー画面300を示す。メニュー画面300は、「タイトルメニュー」と名付けられ、それぞれ「本編再生」、「映像特典」および「音声設定」と表示された３つのボタン301A、301Bおよび301Cが配置される。この図86の例では、「本編再生」ボタン301Aの外枠の色がボタン301Aの元の外枠の色と異なった色にされており、「本編再生」ボタン301Aが選択状態となっていることが示される。 10

#### 【0014】

この状態で、ユーザがリモコンの方向キーを所定に操作することにより、図87A、図87Bおよび図87Cにそれぞれ例示されるように、選択状態のボタンを変更することができる。選択されたボタンは、上述の図86と同様に、外枠の色が非選択状態の場合に対して異ならされる。さらに、例えば図86の状態で、ユーザがリモコンに設けられた決定ボタンを操作すると、その時点で選択状態にある「本編再生」ボタン301Aが図88に一例が示されるように、実行状態を表す色に変化される。その後、メニュー画面300が消え、本編が再生されることになる。以上が、例えばDVDビデオによるボタンの基本動作である。 20

#### 【0015】

ところで、図86に例示されるメニュー画面300は、図89A、図89Bおよび図89Cにそれぞれ一例が示されるように、背景画像310、サブピクチャ311およびハイライト312の３種類のデータから構成される。背景画像310は、静止画や、当該DVDに収録されるコンテンツ本体の動画などが表示される。

#### 【0016】

サブピクチャ311は、図90に例示されるように、１ピクセル当たり２ビットの情報で表現された１枚のビットマップ画像と、４色の色情報（A0、B0、C0、D0）と、当該サブピクチャ311の表示開始位置を示す座標（X、Y）とを有する。色情報A0、B0、C0およびD0は、それぞれ１組のR（赤）G（緑）B（青）データからなる１色の色情報データであって、R、G、B各色について各々８ビットの情報を有する。ビットマップ画像は、１ピクセルにつき２ビットの情報を有し、２ビットを用いて上述の４色の色情報（A0、B0、C0、D0）の中から１色をピクセル毎に選択する。色情報は、透明度情報も含まれており、サブピクチャ311に対して背景画像310が透けて見える領域を設けることができる。サブピクチャ311の表示位置は、例えばサブピクチャ311の左上隅の位置が背景画像310に対する相対的な座標（X、Y）で示される。 30

#### 【0017】

サブピクチャ311は、さらに、表示開始および停止時刻を示す情報、当該サブピクチャ311に対してフェードイン、フェードアウトといった視覚効果を施すコマンドを有することができる。 40

#### 【0018】

DVDビデオにおいては、同時に複数のビットマップ画像を表示することができない。そのため、上述した図86に例示されるような、複数のボタンが配置されたようなメニュー画面300の表示は、図89Bに例示されるような、３つのボタン画像を含む１枚の大きなビットマップ画像によりなされる。図89のサブピクチャ311のビットマップ画像において、ボタン301A、301Bおよび301Cの外側の領域を透明領域として設定することで、このサブピクチャ311と背景画像310とを合成した際に、ボタン301A、301Bおよび301Cの表示領域以外に背景画像310を透過させて表示させることができる。 50

#### 【0019】

ハイライト312は、サブピクチャ311に用いられている4色を、他の4色に置き換えるための情報である。図90に例示されるように、ハイライト312は、色情報として、選択状態の色情報(A1, B1, C1, D1)および実行状態の色情報(A2, B2, C2, D2)を有する。これらの色情報も、上述のサブピクチャ311と同様、RGBそれぞれ8ビットで表現される4色の色情報である。

#### 【0020】

また、ハイライト312は、位置情報として、色を置き換える領域の座標の組を有する。色を置き換える範囲は、サブピクチャ311の画像全体に限らず、サブピクチャ311内の一部の矩形領域とすることができる。このハイライト312により色を変化させるサブピクチャ311内の矩形領域の数がユーザが選択可能なボタンの数に相当する。それぞれの矩形領域の表示位置は、左上隅および右下隅の位置が座標(X, Y)で示される。例えば、ボタン301Aに対応するハイライト312Aの位置は、座標(X1, Y1)および(X1', Y1')で示される。ボタン301Bおよび301Cにそれぞれ対応するハイライト312Bおよび312Cも、同様である。

#### 【0021】

一例として、ハイライト312Aは、背景画像310の座標(X1, Y1)および(X1', Y1')で示される領域の色情報(A0, B0, C0, D0)を、例えば選択状態の色として指定される色情報(A1, B1, C1, D1)に置き換える。このとき、背景画像310の色情報A0がハイライト312Aの色情報A1に置き換えられる。同様に、背景画像310の色情報B0が色情報B1に、色情報C0が色情報C1に、色情報D0が色情報D1に、それぞれ置き換えられる。

#### 【0022】

このハイライト312による色の置き換えの例を、図86、図87および図88を用いて上述した、メニュー画面300におけるボタン301Aの状態に応じた表示変化に対応させて説明する。例えばボタン301Aが非選択状態のときには、ボタン301Aの枠、表面および文字がそれぞれ色情報B0、色情報C0および色情報D0を用いて表示されているものとする。ボタン301Aを選択状態とすると、ボタン301Aの枠色B0がハイライト312Aの選択状態の対応する色情報B1で置き換えられる。表面色C0および文字色D0は、変化しない。次いで、ボタン301Aが実行状態とされると、選択状態の色に対して、さらに、ボタン301Aの表面色C0が色情報C1に置き換えられる。枠色B1および文字色D0は、選択状態に対して変化しない。

#### 【0023】

DVDビデオにおいて、通常の映像再生時は、背景画像310に相当する画像だけが表示されている。また、字幕付きの映画などの場合は、映画本編が再生されている背景画像310と、字幕が表示されているサブピクチャ311とが合成された状態で表示がなされている。

#### 【発明の開示】

#### 【発明が解決しようとする課題】

#### 【0024】

しかしながら、このDVDビデオの場合、サブピクチャ311、選択状態を表すハイライト312、実行状態を表すハイライト312は、それぞれ最大4色しか用いることができず、色数の多いサブピクチャを表示することができなかつたという問題点があった。

#### 【0025】

また、ハイライト312は、サブピクチャ311の色を置き換え変更するだけであつたため、例えば選択状態や実行状態においてボタン中に表示されている文字を変えることや、ボタンの形状が変化するような効果を実現することができず、表現力に乏しかったという問題点があった。

#### 【0026】

さらに、従来では、字幕表示とボタン表示とを、サブピクチャ311を用いた同一の仕組みで実現していたため、字幕とボタンとをそれぞれ独立して表示制御することや、字幕

あるいはボタン表示に透明度を設定してこれらを重ね合わせて表示させるといった、合成処理もできなかったという問題点があった。

【0027】

さらにまた、このDVDビデオの場合、例えばメニュー画面を呼び出した際に、背景に表示される動画データの再生が中断されていた。このように、従来では、例えばインタラクティブな機能が実現されていても、その機能を実現するためのユーザインターフェイスの自由度が低いという問題点があった。

【0028】

また、DVDビデオでは、字幕の表示や変化に同期させて効果音を発生させる仕組みが規格で定められていないため、例えば字幕に同期させて効果音を発生させるようなことができなかったという問題点があった。 10

【0029】

さらにボタンについても、ユーザがボタンを選択状態にした際に発せられる効果音や、ボタンの選択状態時に決定キーを操作するなどして、当該ボタンを実行状態にした際に発せられるクリック音のような効果音を発生させる仕組みが規格で定められていなかったため、表現豊かなユーザインターフェイスを実現することが困難であるという問題点があった。

【0030】

なお、ここでいう効果音とは、動画プレーン上に表示される動画や静止画に同期して再生される音声（例えば、映画などにおいて、映画映像と対になって収録されている音声）とは別途に用意された音声データが、字幕やボタンの表示制御に同期または誘引されて再生される音声のことをいう。 20

【0031】

したがって、この発明の目的は、大容量の再生専用光ディスクにおいて自由度の高いユーザインターフェイスを実現するための再生装置、再生方法、再生プログラムおよび記録媒体を提供することにある。

【0032】

また、この発明の他の目的は、大容量の再生専用光ディスクにおいて、より表現豊かなユーザインターフェイスを実現する再生装置、再生方法、再生プログラムおよび記録媒体を提供することにある。 30

【課題を解決するための手段】

【0033】

請求項1に記載の発明は、上述した課題を解決するために、円盤状記録媒体に記録されたコンテンツデータを入力する再生装置において、記録媒体から再生された、少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画データおよび字幕データを含むリアルタイムストリームとを入力する入力手段と、入力手段により入力されたプログラムコードを格納するコード格納手段と、入力手段により入力された画像データを格納する画像データ格納手段と、入力手段により入力された動画データをデコードした復号動画データと、入力手段により入力された字幕データをデコードした復号字幕データとを合成する第1の合成手段と、コード格納手段に格納されたプログラムコードに基づき、画像データ格納手段に格納された画像データをデコードした復号画像データと、第1の合成手段により合成された動画字幕合成データとを合成する第2の合成手段とを有することを特徴とする再生装置である。 40

【0034】

また、請求項50に記載の発明は、円盤状記録媒体に記録されたコンテンツデータを入力する再生方法において、記録媒体から再生された、少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画データおよび字幕データを含むリアルタイムストリームとを入力する入力のステップと、入力のステップにより入力されたプログラムコードをコード格納手 50

段に格納するステップと、入力ステップにより入力された画像データを画像データ格納手段に格納するステップと、入力ステップにより入力された動画データをデコードした復号動画データと、入力ステップにより入力された字幕データをデコードした復号字幕データとを合成する第1の合成のステップと、コード格納手段に格納されたプログラムコードに基づき、画像データ格納手段に格納された画像データをデコードした復号画像データと、第1の合成のステップにより合成された動画字幕合成データとを合成する第2の合成のステップとを有することを特徴とする再生方法である。

【0035】

また、請求項51に記載の発明は、円盤状記録媒体に記録されたコンテンツデータを再生する再生方法をコンピュータ装置に実行させる再生プログラムにおいて、記録媒体から再生された、少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画データおよび字幕データを含むリアルタイムストリームとを入力する入力ステップと、入力ステップにより入力されたプログラムコードをコード格納手段に格納するステップと、入力ステップにより入力された画像データを画像データ格納手段に格納するステップと、入力ステップにより入力された動画データをデコードした復号動画データと、入力ステップにより入力された字幕データをデコードした復号字幕データとを合成する第1の合成のステップと、コード格納手段に格納されたプログラムコードに基づき、画像データ格納手段に格納された画像データをデコードした復号画像データと、第1の合成のステップにより合成された動画字幕合成データとを合成する第2の合成のステップとを有する再生方法をコンピュータ装置に実行させることを特徴とする再生プログラムである。

【0036】

また、請求項52に記載の発明は、円盤状記録媒体に記録されたコンテンツデータを再生する再生方法をコンピュータ装置に実行させる再生プログラムが記録された記録媒体において、記録媒体から再生された、少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画データおよび字幕データを含むリアルタイムストリームとを入力する入力ステップと、入力ステップにより入力されたプログラムコードをコード格納手段に格納するステップと、入力ステップにより入力された画像データを画像データ格納手段に格納するステップと、入力ステップにより入力された動画データをデコードした復号動画データと、入力ステップにより入力された字幕データをデコードした復号字幕データとを合成する第1の合成のステップと、コード格納手段に格納されたプログラムコードに基づき、画像データ格納手段に格納された画像データをデコードした復号画像データと、第1の合成のステップにより合成された動画字幕合成データとを合成する第2の合成のステップとを有する再生方法をコンピュータ装置に実行させる再生プログラムが記録されたことを特徴とする記録媒体である。

【0037】

また、請求項53に記載の発明は、コンテンツデータが記録された円盤状の形状を有する記録媒体において、少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画データおよび字幕データを含むリアルタイムストリームとが記録され、プログラムコードに基づき、再生後に画像データ格納手段に格納された画像データをデコードした復号画像データと、再生された動画データをデコードした復号動画データおよび再生された字幕データをデコードした復号字幕データが合成された動画字幕合成データとが合成されることを特徴とする記録媒体である。

【0038】

上述したように、請求項1、50、51、52に記載の発明は、記録媒体から再生された、少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画データおよび字幕データを含むリアルタイムストリームとを入力し、入力されたプログラムコードおよび画像データ

をコード格納手段および画像データ格納手段にそれぞれ格納し、入力された動画データおよび字幕データをそれぞれデコードした復号動画データおよび復号字幕データとを合成して動画字幕合成データとし、コード格納手段に格納されたプログラムコードに基づき、画像データ格納手段に格納された画像データをデコードした復号画像データと動画字幕合成データとを合成するようにしているため、再生時に、同一の画像データを用いた操作画面を、異なるタイミングでそれぞれ表示させることが容易である。

【0039】

また、請求項53に記載の発明は、少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画データおよび字幕データを含むリアルタイムストリームとが記録され、プログラムコードに基づき、再生後に画像データ格納手段に格納された画像データをデコードした復号画像データと、再生された動画データをデコードした復号動画データおよび再生された字幕データをデコードした復号字幕データが合成された動画字幕合成データとが合成されるようにしているため、再生時に、同一の画像データを用いた操作画面を、異なるタイミングでそれぞれ表示させることが容易である。

【発明の効果】

【0040】

この発明によれば、記録済み大容量ディスクにおいて、動画を表示する動画プレーンと、字幕を表示する字幕プレーンと、メニュー画面などインタラクティブな機能を有する画面を表示するグラフィクスプレーンとがそれぞれ独立して設けられ、これらのプレーンが合成されて表示されるようにしているため、動画プレーンに動画を表示させ、その動画を背景としてグラフィクスプレーンにメニュー画面などを表示させることができるという効果がある。

【0041】

また、この発明によれば、グラフィクスプレーンに表示するための画像データを格納するためのバッファが設けられているため、グラフィクスプレーンの表示を同一の画像データを繰り返し用いて表示することができる。これにより、メニュー画面などをより高い自由度で構成することができるという効果がある。

【0042】

さらに、この発明では、グラフィクスプレーンに表示されるグラフィクスの表示制御を、表示制御命令を用いて記述しているため、インタラクティブな機能をグラフィクスプレーンに表示される画面を用いて実現することができるという効果がある。

【0043】

さらにまた、この発明によれば、記録済み大容量ディスクにおいて、動画を表示する動画プレーンと、字幕を表示する字幕プレーンと、メニュー画面などインタラクティブな機能を有する画面を表示するグラフィクスプレーンとがそれぞれ独立して設けられ、これらのプレーンが合成されて表示されるようにしている。そして、字幕プレーンとグラフィクスプレーンに表示するオブジェクトの形式として共通のグラフィクスオブジェクトを定義し、デコーダモデル、表示制御命令と動作モデルを定義した。これにより、動画に同期して字幕やボタンを表示することが可能となる効果がある。

【0044】

また、字幕やボタンのスクロール表示、移動表示といった簡単なアニメーションや、ユーザ入力に応じて画像の内容が変化する表現力のあるボタンを実現することができるという効果がある。

【0045】

さらに、この発明では、グラフィクスオブジェクトに音声データを含め、グラフィクスオブジェクトに含まれるボタン画像の表示と同時に音声データを再生するデコーダモデルを定義した。そのため、グラフィクスオブジェクトの表示に同期した音声データの再生を容易に行うことができるという効果がある。

【0046】

さらにまた、この発明では、グラフィクスオブジェクトの表示制御命令に対して、音声データの再生を指示する命令を定義した。また、グラフィクスオブジェクトの表示制御命令に対して、グラフィクスオブジェクトに含まれる画像データに対して音声データを割り当てることができるようにしたため、任意の時刻での効果音などの音声データの再生や、効果音を伴う字幕やボタンの実現などが可能となる効果がある。

#### 【0047】

また、この発明では、GUI (Graphical User Interface)により表現されるメニュー画面上に表示されるボタンについて、ユーザがボタンを選択状態にした際や、ボタンの選択状態時に決定キーを操作するなどしてボタンを実行状態とした際に発せられる、クリック音のような効果音を発生させる仕組みを提供している。そのため、ブルーレイ (Blu-ray) ディスクといった、大容量の記録媒体に予め記録されたプログラムに対する、ユーザによるインタラクティブな操作を可能とする再生装置、再生方法、再生プログラムおよび記録媒体において、表現豊かなユーザインターフェイスを実現することができる効果がある。

【発明を実施するための最良の形態】

#### 【0048】

以下、この発明について、下記の構成に従い説明する。

1. BD A Vフォーマットの概要
2. 発明の実施の第1の形態
  - 2-1. プレーンについて
  - 2-2. メニュー画面
  - 2-3. シナリオについて
  - 2-4. 仮想プレーヤモデルについて
  - 2-5. メソッドについて
  - 2-6. シナリオの実行について
  - 2-7. デコーダモデル
  - 2-8. ユーザインターフェイス
  - 2-9. データ構造について
  - 2-10. グラフィクスオブジェクトのデコーダモデルについて
  - 2-11. グラフィクスの転送速度について
  - 2-12. グラフィクスオブジェクトの構造の詳細
  - 2-13. 効果音について
  - 2-14. 音声データのディスクへの格納方法について
    - 2-14 a. 第1の格納方法
    - 2-14 b. 第2の格納方法
    - 2-14 c. 第3の格納方法
  - 2-15. プレーンの別の例について
  - 2-16. デコーダモデルの別の例について
  - 2-17. デコーダモデルのさらに別の例について
3. 発明の実施の第2の形態
  - 3-1. シナリオについて
  - 3-2. シナリオの分類
  - 3-3. 仮想プレーヤモデルについて
  - 3-4. シンタクスについて
  - 3-5. コマンドについて
4. その他

#### 【0049】

1. BD A Vフォーマットの概要

先ず、理解を容易とするために、Blu-ray Discに関し、"Blu-ray Disc Rewritable Format Ver1.0 part3 Audio Visual Specifications"で規定されている、コン

テンツすなわちディスクに記録されたA V (Audio/Video)データの管理構造について説明する。以下では、この管理構造をB D A Vフォーマットと称する。

【0050】

例えばM P E G (Moving Pictures Experts Group)ビデオやM P E G オーディオなどの符号化方式で符号化され、M P E G 2システムに従い多重化されたビットストリームは、クリップA Vストリーム（またはA Vストリーム）と称される。クリップA Vストリームは、B l u - r a y D i s cに関する規格の一つである"Blu-ray Disc Rewritable Format Ver1.0 part2"で定義されたファイルシステムにより、ファイルとしてディスクに記録される。このファイルを、クリップA Vストリームファイル（またはA Vストリームファイル）と称する。

10

【0051】

クリップA Vストリームファイルは、ファイルシステム上での管理単位であり、ユーザにとって必ずしも分かりやすい管理単位であるとは限らない。ユーザの利便性を考えた場合、複数のクリップA Vストリームファイルに分割された映像コンテンツを一つにまとめて再生する仕組みや、クリップA Vストリームファイルの一部だけを再生する仕組み、さらには、特殊再生や頭出し再生を滑らかに行うための情報などをデータベースとしてディスクに記録しておく必要がある。B l u - r a y D i s cに関する規格の一つである"B l u - r a y D i s c Rewritable Format Ver1.0 part3"で、このデータベースが規定される。

【0052】

図1は、A Vストリームファイルの一部または全部の範囲を指定して、必要な部分だけを並べて再生する再生順序指定の仕組みを模式的に示す。図1において、プレイリスト(P l a y L i s t)は、A Vストリームファイルの一部または全部の範囲を指定して、必要な部分だけを再生する指示を与える。ユーザがコンテンツの再生を行う際には、このプレイリストの単位で選択する。つまり、プレイリストは、ユーザからみてひとまとまり、すなわち、再生が連続に行われることをユーザが暗黙に期待する、映像・音声の単位である。

【0053】

最も簡単なプレイリストの構成は、記録が開始されてから記録が終了されるまでの一繋がりA Vストリームファイルからなり、編集をしなければ、これが一つのプレイリストになる。

【0054】

プレイリストは、再生するA Vストリームファイルの指定と、指定されたA Vストリームファイルの再生箇所を指定する再生開始点と再生終了点の集まりとから構成される。この再生開始点と再生終了点の情報を一組としたものは、プレイアイテム(P l a y I t e m)と称される。プレイリストは、プレイアイテムの集合で構成される。プレイアイテムを再生することは、そのプレイアイテムに参照されるA Vストリームファイルの一部分を再生することになる。

30

【0055】

クリップA Vストリームは、上述したように、ビデオデータやオーディオデータがM P E G 2 T S (トランスポートストリーム)の形式などに多重化されたビットストリームである。このクリップA Vストリームに関する情報がクリップ情報(C l i p I n f o r m a t i o n)としてファイルに記録される。

40

【0056】

クリップA Vストリームファイルと、対応するクリップ情報が記録されたクリップ情報ファイルとをひとまとまりのオブジェクトと見なし、クリップ(C l i p)と称する。すなわち、クリップは、クリップA Vストリームとクリップ情報とから構成される、一つのオブジェクトである。

【0057】

ファイルは、一般的に、バイト列として扱われる。クリップA Vストリームファイルのコンテンツは、時間軸上に展開され、クリップ中のエン트리ポイントは、主に時間ベースで指定される。所定のクリップへのアクセスポイントのタイムスタンプが与えられた場

50



合、クリップAVストリームファイルの中でデータの読み出しを開始すべきアドレス情報を見つけるために、クリップ情報ファイルを用いることができる。

#### 【0058】

1のディスク中に記録された全てのプレイリストおよびクリップは、ボリュームインフォメーション(Volume Information)で管理される。

#### 【0059】

図2は、上述のようなクリップAVストリーム、クリップ情報(Stream Attributes)、クリップ、プレイアイテムおよびプレイリストの関係を示すUML(Unified Modeling Language)図である。プレイリストは、1または複数のプレイアイテムに対応付けられ、プレイアイテムは、1のクリップに対応付けられる。1のクリップに対して、それぞれ開始点および/または終了点異なる複数のプレイアイテムに対応付けることができる。1のクリップから1のクリップAVストリームファイルが参照される。同様に、1のクリップから1のクリップ情報ファイルが参照される。また、クリップAVストリームファイルとクリップ情報ファイルとは、1対1の対応関係を有する。このような構造を定義することにより、クリップAVストリームファイルを変更することなく、任意の部分だけを再生する、非破壊の再生順序指定を行うことが可能となる。

#### 【0060】

また、図3のように、複数のプレイリストから同一のクリップを参照することもできる。図3の例では、クリップ1が2つのプレイリスト2および3から参照されている。図3において、クリップ1は、横方向が時間軸を表す。プレイリスト2により、コマーストリー1の区間a~fが参照される。一方、プレイリスト3により、シーンeを含むクリップ1の区間d~gが参照される。プレイリスト2を指定することで、クリップ1の区間a~fを再生することができ、プレイリスト3を指定することで、クリップ1の区間d~gを再生することができる。

#### 【0061】

次に、"Blu-ray Disc Rewritable Format Ver1.0 part3"で規定された、記録媒体に記録されるファイルの管理構造について、図4を用いて説明する。ファイルは、ディレクトリ構造により階層的に管理される。記録媒体上には、先ず、1つのディレクトリ(図4の例ではルートディレクトリ)が作成される。このディレクトリの下が、1つの記録再生システムで管理される範囲とする。

#### 【0062】

ルートディレクトリの配下に、ディレクトリBD-AVが置かれる。ディレクトリBD-AVは、図4のように、ディレクトリBD-AV、BD-AV1、BD-AV2、・・・、BD-AVnの如く、ルートディレクトリの配下に複数置くことができる。以下では、複数のディレクトリBD-AV、BD-AV1、BD-AV2、・・・、BD-AVnをディレクトリBD-AVで代表させて説明する。

#### 【0063】

ディレクトリBD-AVの配下には、以下の6種類のファイルが置かれる。

- (1) info.bdav
- (2) menu.tidx, mark.tidx
- (3) menu.tdt1, menu.tdt2, mark.tdt1, mark.tdt2
- (4) #####.rp1s, #####.vp1s
- (5) %%%.clpi
- (6) \*\*\*\*\*.m2ts

#### 【0064】

なお、(4)のファイル「#####.rp1s」および「#####.vp1s」において、「#####」は、任意の番号を示す。同様に、(5)のファイル「%%%.clpi」において、「%%%」は任意の番号を示す。また、(6)のファイル「\*\*\*\*\*.m2ts」において、「\*\*\*\*\*」は、ファイル「\*\*\*\*\*.m2ts」がファイル「%%%.clpi」と一対一で対応するような番号である。例えば、番号「\*\*\*\*\*」は、番号「%%%」と同一とすることができる。



## 【0065】

(1) のファイル「info.bdav」は、ディレクトリBD A V全体の情報が格納されるファイルである。(2) のファイル「menu.tidx」および「mark.tidx」は、サムネイル画像の情報が格納されるファイルである。(3) のファイル「menu.tdt1」、「menu.tdt2」、「mark.tdt1」および「mark.tdt2」は、サムネイル画像が格納されるファイルである。各ファイルの拡張子「tdt1」および「tdt2」は、当該ファイルに格納されているサムネイル画像データが暗号化されているか否かを示す。

## 【0066】

(4) のファイル「#####.rp1s」、「#####.vp1s」は、プレイリストの情報が格納されるファイルである。ファイル「#####.rp1s」、「#####.vp1s」は、ディレクトリBD A V 10の配下に設けられたディレクトリPLAYLISTのさらに配下に置かれている。

## 【0067】

(5) のファイル「%%%%.clpi」は、クリップ情報が格納されるファイルである。ファイル「%%%%.clpi」は、ディレクトリBD A Vの配下に設けられたディレクトリCLIPINFのさらに配下に置かれている。(6) のファイル「\*\*\*\*\*.m2ts」は、クリップAVストリームが格納される、クリップAVストリームファイルである。このクリップAVストリームファイルは、ファイル名の番号「\*\*\*\*\*」により、1つのクリップ情報ファイル「%%%%.clpi」に対応付けられる。ファイル「\*\*\*\*\*.m2ts」は、ディレクトリBD A Vの配下に設けられたディレクトリSTREAMのさらに配下に置かれる。

## 【0068】

各ファイルについて、より詳細に説明する。(1) のファイル「info.bdav」は、ディレクトリBD A Vの配下に唯一、設けられる。図5は、ファイル「info.bdav」の一例の構造を表すシンタクスを示す。ここでは、シンタクスをコンピュータ装置などのプログラムの記述言語として用いられるC言語の記述法に基づき示す。これは、他のシンタクスを表す図において、同様である。

## 【0069】

図5において、ファイル「info.bdav」の内部は、機能別の情報毎にブロックが構成される。フィールドtype\_indicatorは、文字列「BD A V」が格納され、このファイルがファイル「info.bdav」であることが示される。フィールドversion\_numberは、このファイル「info.bdav」のバージョンが示される。ブロックUIAppInfoBD A V Oは、ディレクトリBD A V 30の配下に関する情報が格納される。ブロックTableOfPlayLists()は、プレイリストの並びに関する情報が格納される。ブロックMakersPrivateData()は、記録再生装置のメーカー固有の情報が格納される。

## 【0070】

ファイル「info.bdav」の先頭には、各ブロックの先頭を表すアドレスが記述される。例えば、フィールドTableOfPlayLists\_Start\_addressは、ブロック「TableOfPlayLists()」の開始する位置がファイル内の相対バイト数で示される。

## 【0071】

図6は、ブロックUIAppInfoBD A V Oの一例の構造を表すシンタクスを示す。フィールドlengthは、フィールドlength直後のフィールドからこのブロックUIAppInfoBD A V Oの最後ま 40での長さがバイトで示される。フィールドBD A V\_character\_setは、このブロックUIAppInfoBD A V O内のフィールドBD A V\_nameに記録されている文字列の文字セットが示される。文字セットとしては、ASCII、Unicodeなどが選択できる。

## 【0072】

フラグBD A V\_protect\_flagは、ディレクトリBD A Vの配下に置かれるコンテンツを、ユーザに制限無しに見せて良いか否かを示す。例えば、このフラグが値「1」にセットされており、且つ、ユーザが正しく暗証番号PIN(Personal Identification Number)を入力できた場合に、ディレクトリBD A Vの配下に置かれたコンテンツをユーザに見せることができるものとする。一方、フラグBD A V\_protect\_flagが値「0」にセットされている場合には、ユーザによる暗証番号PINの入力が無くても、ディレクトリBD A Vの配下に置 50

かれたコンテンツをユーザに見せることができる。

【0073】

暗証番号PINは、フィールドPINに記録される。暗証番号PINは、例えば4桁の0～9の数字からなり、上述のように再生制御を有効にした場合に必要となる暗証番号を表す。暗証番号PINのそれぞれの数字は、例えばISO(International Organization for Standardization)/IEC(International Electrotechnical Commission)646の規定に従って符号化される。

【0074】

このブロックUIAppInfoBDV()に示される以上の情報により、ディレクトリBDVに対する再生制限が規定される。なお、詳細は後述するが、個々のプレイリストに対する再生制限については、ファイル「#####.rp1s」、「#####.vp1s」内に示されるブロックUIAppInfoPlayList()の中で定義されるフラグplayback\_control\_flagによって規定される。

【0075】

この例では、ディレクトリBDVの配下のコンテンツの再生を再開する際に、優先して再生すべきプレイリストを指定する、レジューム機能を用いることができる。このレジューム機能は、前回視聴を中断した箇所から再生を再開したい場合などに使うことを想定している。

【0076】

図6において、フラグresume\_valid\_flagは、このレジューム機能の有効/無効を示す。例えば、このフラグが値「0」にセットされている場合には、レジューム機能は無効とされ、このフラグが値「1」にセットされている場合には、レジューム機能が有効とされ、その際、フィールドresume\_PlayList\_file\_nameで指定されるプレイリストを、優先して再生すべきプレイリストとして扱う。

【0077】

フィールドref\_to\_menu\_thumbnail\_indexは、このディレクトリBDVを代表するサムネイル画像が存在する場合、そのサムネイル画像を特定するサムネイル番号が格納される領域である。Blu-ray Discの規格では、ディレクトリBDVを代表する静止画を特にメニューサムネイルと呼ぶ。フィールドref\_to\_menu\_thumbnail\_indexで指定されるインデックスthumbnail\_indexを持つサムネイル画像がこのディレクトリBDVのメニューサムネイルとなる。

【0078】

フィールドBDV\_name\_lengthは、フィールドBDV\_nameに示されるディレクトリBDV名バイト長を示す。このフィールドBDV\_nameにおいて、左から、フィールドBDV\_name\_lengthで示されるだけのバイト数が有効な文字列であり、それがこのディレクトリBDVの名前を示す。なお、フィールドBDV\_nameの中で、フィールドBDV\_name\_lengthで示された有効な文字列の後のバイト列には、どのような値が入っていても良い。

【0079】

図7は、ブロックTableOfPlayLists()の一例の構造を表すシンタクスを示す。フィールドnumber\_of\_PlayListsは、このディレクトリBDVの配下に置かれたプレイリストの数が示される。このプレイリスト数をループ変数とする、次に続く「for」文のループにおいてフィールドPlayList\_file\_nameにより示されるプレイリストの並びが、プレイリスト一覧表示画面などに表示される際のプレイリストの順序を示す。プレイリストは、フィールドPlayList\_file\_nameにより、「#####.rp1s」や「#####.vp1s」といったファイル名で指定される。

【0080】

ファイル「#####.rp1s」および「#####.vp1s」は、上述したように、ディレクトリPLAYLISTの配下に置かれる。これらのファイルは、1つのプレイリストと一対一に対応している。

【0081】

図8は、ファイル「#####.rp1s」および「#####.vp1s」の一例の構造を表すシンタクス

を示す。図8において、ファイル「#####.rpls」および「#####.vpls」の内部は、機能別の情報毎にブロックが構成される。フィールドtype\_indicatorにこのファイルを示す文字列が格納され、フィールドversion\_numberにこのファイルのバージョンが示される。

【0082】

ブロックUIAppInfoPlaylist()は、このプレイリストに関する属性情報が格納される。ブロックPlaylist()は、このプレイリストを構成するプレイアイテムに関する情報が格納される。ブロックPlaylistMark()は、このプレイリストに付されるマークの情報が格納される。ブロックMakersPrivateData()は、このプレイリストファイルを記録した装置の、メーカ固有の情報が格納される。ファイル「#####.rpls」および「#####.vpls」の先頭に設けられるフィールドPlaylist\_start\_address、PlaylistMark\_start\_addressおよびMakersPrivateData\_start\_addressには、それぞれ対応するブロックの先頭アドレスが例えば32ビットのアドレス情報として示される。

【0083】

各ブロックの先頭アドレスがファイル「#####.rpls」および「#####.vpls」の先頭に示されるために、ブロックの前および／または後ろに、任意の長さのデータpadding\_wordを挿入することができる。但し、ファイル「#####.rpls」および「#####.vpls」の最初のブロックであるブロックUIAppInfoPlaylist()の開始位置は、ファイルの先頭から320バイト目に固定される。

【0084】

図9は、ブロックUIAppInfoPlaylist()の一例の構造を表すシンタクスを示す。ブロックUIAppInfoPlaylist()は、プレイリストの再生において直接的に必要とされない、プレイリストに関する各種の属性情報が格納される。フィールドPlaylist\_character\_setは、プレイリストに関する文字列情報の文字セットが指定される。

【0085】

フラグplayback\_control\_flagは、情報表示やプレイリストの再生を暗証番号PINに基づき制限するか否かが指定される。例えばフラグplayback\_control\_flagが値「1」であれば、ユーザにより正しい暗証番号PINが入力されなければ、プレイリストのサムネイル画像などの情報表示や、プレイリストの再生ができないとされる。フラグwrite\_protect\_flagは、消去禁止フラグである。例えばこのフラグwrite\_protect\_flagが値「1」であれば、ユーザが容易にプレイリストを消去できないようなユーザインタフェースにする必要がある。フラグis\_played\_flagは、このプレイリストが再生済みであることを示す。フラグis\_edited\_flagは、このプレイリストが編集されたことを示す。

【0086】

フィールドtime\_zoneは、このプレイリストが記録された地点のタイムゾーンを示す。フィールドrecord\_time\_and\_dateは、プレイリストの記録日時を示す。フィールドPlaylist\_durationは、プレイリストの再生時間を示す。

【0087】

フィールドmaker\_IDおよびmaker\_model\_codeは、このプレイリストを最終更新した記録機のメーカおよび記録機をそれぞれ特定する情報が例えば番号で示される。フィールドchannel\_numberは、記録したクリップAVストリームのチャンネル番号が示され、フィールドchannel\_nameは、当該チャンネル名が示される。フィールドchannel\_nameに示されるチャンネル名の長さがフィールドchannel\_name\_lengthに示される。フィールドchannel\_nameの領域のうち、フィールドchannel\_name\_lengthで示される長さの文字列が有効となる。フィールドPlaylist\_nameは、フィールドPlaylist\_name\_lengthに示される値を有効長としてプレイリスト名が示される。フィールドPlaylist\_detailは、フィールドPlaylist\_detail\_lengthに示される値を有効長としてプレイリストの詳細情報が示される。

【0088】

図10は、ブロックPlaylist()の一例の構造を表すシンタクスを示す。フィールドlengthは、フィールドlengthの直後のフィールドからこのブロックPlaylist()の終端までのバイト長を示す。フィールドPL\_CPI\_typeは、このプレイリストが持つCPI (Characteris 50

tic Point Information: 特徴点情報) の種類を示す。フィールド `number_of_PlayItems` は、このプレイリストを構成するプレイアイテムの数を示す。フィールド `number_of_SubPlayItems` は、このプレイリストに付けられているアフレコオーディオ用のプレイアイテム (サブプレイアイテム) の数を示す。詳細は省略するが、プレイリストは、所定の条件を満たす場合にだけ、サブプレイアイテムを持つことができる。

【0089】

ブロック `PlayItem()` は、プレイアイテムの情報が格納される。ブロック `SubPlayItem()` は、サブプレイアイテムの情報が格納される。

【0090】

図11は、ブロック `PlayItem()` の一例の構造を表すシンタクスを示す。フィールド `Clip_Information_file_name` は、このプレイアイテムが参照しているクリップと一対一に対応するクリップ情報ファイル (拡張子が `clpi` であるファイル) のファイル名が文字列で示される。クリップ情報ファイルは、拡張子が `「clpi」` とされているファイルである。

【0091】

フィールド `Clip_codec_identifier` は、このプレイアイテムにより参照されるクリップの符号化方式を示す。この例では、フィールド `Clip_codec_Identifier` は、値 `「M2TS」` に固定的とされる。フィールド `connection_condition` は、このプレイアイテムが次のプレイアイテムとの間でどのような接続がされているかを示す情報である。すなわち、フィールド `connection_condition` により、プレイアイテムとプレイアイテムとの間が継ぎ目なくシームレスに再生できるか否かが示される。

【0092】

フィールド `ref_to_STC_id` は、このプレイアイテムにより参照されるクリップ内のシーケンス `STC_sequence` を指定する。シーケンス `STC_sequence` は、`MPEG2 TS (Transport Stream)` における時間軸の基準である `PCR (Program Clock Reference)` が連続な範囲を表す `Blue-ray Disc` 規格独自の構造である。シーケンス `STC_sequence` には、クリップ内で一意な番号 `STC_id` が割り当てられる。このシーケンス `STC_sequence` 内では、不連続の無い一貫した時間軸を定義できるので、プレイアイテムの開始時刻および終了時刻を一意に定めることができる。つまり、各プレイアイテムの開始点と終了点は、同一のシーケンス `STC_sequence` に存在していなければならない。フィールド `ref_to_STC_id` では、番号 `STC_id` によりシーケンス `STC_sequence` が指定される。

【0093】

フィールド `IN_time` および `OUT_Time` は、このプレイアイテムにおける開始点および終了点の、シーケンス `STC_sequence` 上でのタイムスタンプ `pts (presentation_time_stamp)` をそれぞれ示す。

【0094】

ブロック `BridgeSequenceInfo()` は、ブリッジクリップ (`Bridge-Clip`) に関する情報が格納される。ブリッジクリップは、図12に一例が示されるように、プレイアイテム間を継ぎ目無く、シームレスに再生する機能を実現する際に作成されるビットストリームである。前のプレイアイテムと現在のプレイアイテムとの継ぎ目の部分で、本来再生すべきビットストリームとは異なるビットストリーム、すなわちブリッジクリップを代わりに再生することで、2つのプレイアイテムをシームレスに再生することができる。このブリッジクリップによる機能は、この発明と関連性が低いので、詳細は省略する。

【0095】

図13は、ブロック `PlaylistMark()` の一例の構造を表すシンタクスを示す。ブロック `PlaylistMark()` は、マークの情報が格納されるデータ構造である。マークは、プレイリスト上の時刻を保持するための構造であって、マークを用いることにより、プレイリスト中への頭出し点の設定、プレイリストを細分化するチャプタの構造を設ける、などの機能が実現される。さらに、後述するグラフィクスプレーン上に描画された画像の表示開始/停止のタイミングを、マークを用いて指定することができる。

【0096】

フィールドlengthは、フィールドlengthの直後のフィールドからブロックPlayListMark Oの終端までのバイト長を示す。フィールドnumber\_of\_PlayList\_marksは、プレイリスト中のマークの数を表す。「for」文での1回のループが1つのマークの情報を示す。フラグmark\_invalid\_flagは、このマークが有効であるか否かを示す。例えば、フラグmark\_invalid\_flagが値「0」でそのマークが有効であることが示され、値「1」であれば、そのマークの情報がデータベース上に存在するが、当該マークがユーザには見えない無効マークとされていることを示す。

#### 【0097】

フィールドmark\_typeは、このマークの種類を表す。マークの種類としては、プレイリストのサムネイル画像（代表画）とする映像の位置を示すマーク、どこまで再生したかを示すレジュームマーク、頭出し点を示すチャプタマーク、ある区間の再生を飛ばして先に進むことを示すスキップマーク、グラフィクス画像の読み込み開始タイミングを示すマーク、グラフィクス画像の表示開始のタイミングを示すマーク、グラフィクス画像の表示停止のタイミングを示すマークなどがある。

#### 【0098】

フィールドmark\_name\_lengthは、後述するフィールドmark\_nameのデータ長を示す。フィールドmaker\_IDは、このマークを作成した記録機の製造者を示す。これは、製造者独自のマークを識別する際に用いられる。フィールドref\_to\_PlayItem\_idは、このマークにより指示される時刻がどのプレイアイテム上に存在するかが示される。フィールドmark\_time\_stampは、このマークにより指示される時刻が示される。

#### 【0099】

フィールドentry\_ES\_PIDは、このマークがどのエレメンタリストリームすなわち映像データおよび／または音声データが符号化されたストリームに対して付与されたものかが示される。フィールドref\_to\_menu\_thumbnail\_indexおよびフィールドref\_to\_mark\_thumbnail\_indexは、マークを視覚的に表すサムネイル画像を示す。サムネイル画像としては、例えばそのマークにより指定される時刻の映像を静止画として抜き出した画像が考えられる。

#### 【0100】

フィールドdurationは、マークが時間軸上で長さを持つ場合に使われる。例えばスキップマークであれば、どのくらいの時間ジャンプをするかがフィールドdurationにより指定される。

#### 【0101】

フィールドmakers\_informationは、製造者独自の情報を記録するための領域である。フィールドmark\_nameは、マークに名前をつけた際、その名前を格納するための領域であり、サイズが上述のフィールドmark\_name\_lengthで指定される。

#### 【0102】

図14は、ファイル「%%%%.clpi」の一例の構造を表すシンタクスを示す。上述したように、ファイル「%%%%.clpi」は、ディレクトリCLIPINFの配下に置かれ、各AVストリームファイル（ファイル「\*\*\*\*\*.m2ts」）毎に作られる。ファイル「%%%%.clpi」の内部は、機能別の情報毎にブロックが構成される。フィールドtype\_indicatorはこのファイルを示す文字列が格納され、フィールドversion\_numberはこのファイルのバージョンが示される。

#### 【0103】

ブロックClipInfo()は、クリップに関する情報が格納される。ブロックSequenceInfo()は、MPEG2システムにおけるトランスポートストリームの時刻基準を表すPCRの不連続点に関する情報が格納される。ブロックProgramInfo()は、MPEG2システムのプログラムに関する情報が格納される。ブロックCPIOは、ランダムアクセス開始可能点などの、AVストリーム中の特徴的な箇所を表す特徴点情報C.P.I.に関する情報が格納される。ブロックClipMark()は、クリップに付された頭出しのためのインデックス点やコマシャルの開始および／または終了点などのマーク情報が格納される。ブロックMakersPriv

ateData()は、記録機の製造者独自の情報が格納される。

【0104】

また、これら各ブロックのファイル「%%%%.clpi」中での先頭を表すアドレス情報が、当該ファイルの先頭部分にフィールドSequenceInfo\_start\_address、ProgramInfo\_start\_address、CPI\_start\_address、ClipMark\_start\_addressおよびMakersPrivateData\_start\_addressとして示される。なお、このクリップファイル「%%%%.clpi」自体は、この発明と関連性が低いので、詳細な説明を省略する。

【0105】

BDAVフォーマットは、以上に概略的に説明したようなデータ構造を持つことにより、クリップAVストリーム中の再生したい部分を開始および終了点の組で指定したプレイアイテムの並びでプレイリストを構築し、ユーザが認識するひとまとまりの再生単位を管理することが可能となる。

【0106】

2. 発明の実施の第1の形態

次に、この発明の実施の第1の形態について説明する。この発明では、上述したBDAVフォーマットを拡張し、リアルタイムデコード方式を用いる動画データおよび字幕データと、蓄積型のデコード方式を用いる画像データとをそれぞれ独立的に用意し、これらのデータを合成して表示することにより、より自由度の高いユーザインターフェイスの実現を可能としている。

【0107】

また、BDAVフォーマットの拡張は、“Blu-ray Disc Rewritable Format Ver1.0 part3”で規定されたプレイリストと互換性が保たれるようになされ、インタラクティブ機能の拡張を実現する。以下、拡張されたBDAVフォーマットを、BDMVフォーマットと称する。このBDMVフォーマットは、再生専用ディスク（BD-ROM：Blu-ray Disc-Read Only Memory）に用いて好適なものである。

【0108】

2-1. プレーンについて

この発明の実施の第1の形態では、画像の表示系について、図15に一例が示されるようなプレーン構成を取る。動画プレーン10は、最も後ろ側（ボトム）に表示され、プレイリストで指定された画像（主に動画データ）が扱われる。字幕プレーン11は、動画プレーン10の上に表示され、動画再生中に表示される字幕データが扱われる。グラフィクスプレーン12は、最も前面に表示され、メニュー画面を表示するための文字データやビットマップデータなどのグラフィクスデータが扱われる。1つの表示画面は、これら3つのプレーンが合成されて表示される。

【0109】

ここで、従来のDVDビデオと異なる特徴的な点は、従来の技術で説明した字幕やメニュー画面、ボタンなどが表示されるサブピクチャを、字幕プレーン11およびグラフィクスプレーン12とに分離し、字幕とボタンとをそれぞれ独立的に制御できるようにしたことである。上述したように、従来のDVDビデオでは、メニュー画面やボタンなどのグラフィクス表示と、字幕表示とを、同じ仕組みで制御しており、これらを同一プレーン上に表示していた。そして、同時に表示できるビットマップ画像は、1枚に限られていた。そのため、従来のDVDビデオでは、複数のビットマップ画像を同時に表示することができなかった。この発明では、字幕を表示する字幕プレーン11と、グラフィクス表示を行うグラフィクスプレーン12とをそれぞれ独立的に設けることで、この従来のDVDビデオの持つ問題点を解消している。

【0110】

なお、字幕プレーン11およびグラフィクスプレーン12が、従来の“Blu-ray Disc Rewritable Format Ver1.0 part3”に対する拡張部分であると考えることができる。

【0111】

動画プレーン10、字幕プレーン11およびグラフィクスプレーン12は、それぞれ独

立して表示が可能とされ、例えば、図16に一例が示されるような解像度および表示可能色を有する。動画プレーン10は、解像度が1920画素×1080ラインで1画素当たり換算したデータ長が16ビットであって、輝度信号Y、色差信号Cb、Crが4:2:2のシステム（以下、YCbCr(4:2:2)）とされる。なお、YCbCr(4:2:2)は、各画素当たり輝度信号Yが8ビット、色差信号Cb、Crがそれぞれ8ビットで、色差信号Cb、Crが水平2画素で一つの色データを構成すると見なすカラーシステムである。

#### 【0112】

グラフィクスプレーン12は、解像度が1920画素×1080ラインで各画素のサンプリング深さが24ビットとされ、カラーシステムは、YCbCr(4:4:4)、または、R(赤色)G(緑色)B(青色)が4:4:4のシステム（以下、RGB(4:4:4)）を選択可能とされる。字幕プレーン11は、1920画素×1080ラインで各画素のサンプリング深さが8ビットとされ、カラーシステムは、256色のパレットを用いた8ビットカラーマップアドレスとされる。

#### 【0113】

グラフィクスプレーン12および字幕プレーン11は、256段階のアルファブレンディングが可能とされており、他のプレーンとの合成の際に、不透明度を256段階で設定することが可能とされている。不透明度の設定は、画素毎に行うことができる。以下では、不透明度 $\alpha$ が( $0 \leq \alpha \leq 1$ )の範囲で表され、不透明度 $\alpha = 0$ で完全に透明、不透明度 $\alpha = 1$ で完全に不透明であるものとする。

#### 【0114】

字幕プレーン11では、例えばPNG(Portable Network Graphics)形式の画像データが扱われる。また、グラフィクスプレーン12でも、例えばPNG形式の画像データを扱うことができる。PNG形式は、1画素のサンプリング深さが1ビット~16ビットとされ、サンプリング深さが8ビットまたは16ビットの場合に、アルファチャンネル、すなわち、それぞれの画素成分の不透明度情報（アルファデータと称する）を付加することができる。サンプリング深さが8ビットの場合には、256段階で不透明度を指定することができる。このアルファチャンネルによる不透明度情報を用いてアルファブレンディングが行われる。また、256色までのパレットイメージを用いることができ、予め用意されたパレットの何番目の要素（インデックス）であるかがインデックス番号により表現される。

#### 【0115】

なお、字幕プレーン11およびグラフィクスプレーン12で扱われる画像データは、PNG形式に限定されない。JPEG方式など他の圧縮符号化方式で圧縮符号化された画像データや、ランレングス圧縮された画像データ、圧縮符号化がなされていないビットマップデータなどを扱うようにしてもよい。

#### 【0116】

図17は、上述の図15および図16に従い3つのプレーンを合成する一例の構成を示す。動画プレーン10の動画データが422/444変換回路20に供給される。動画データは、422/444変換回路20でカラーシステムがYCbCr(4:2:2)からYCbCr(4:4:4)に変換され、乗算器21に入力される。

#### 【0117】

字幕プレーン11の画像データがパレット22に入力され、RGB(4:4:4)の画像データとして出力される。この画像データに対してアルファブレンディングによる不透明度が指定されている場合には、指定された不透明度 $\alpha$  ( $0 \leq \alpha \leq 1$ )がパレット22から出力される。

#### 【0118】

図18は、パレット22の入出力データの一例を示す。パレット22は、例えばPNG形式のファイルに対応したパレット情報がテーブルとして格納される。パレット22は、入力された8ビットの画素データをアドレスとして、インデックス番号が参照される。こ



のインデックス番号に基づき、それぞれ8ビットのデータからなるRGB(4:4:4)のデータが出力される。それと共に、パレット22では、不透明度を表すアルファチャンネルのデータ $\alpha$ が取り出される。

#### 【0119】

図19は、パレット22に格納される一例のパレットテーブルを示す。256個のカラーインデックス値[0x00]~[0xFF]([0x]は16進表記であることを示す)のそれぞれに対して、各々8ビットで表現される三原色の値R、GおよびBと、不透明度 $\alpha$ とが割り当てられる。パレット22は、入力されたPNG形式の画像データに基づきパレットテーブルが参照され、画像データにより指定されたインデックス値に対応する、それぞれ8ビットのデータからなるR、GおよびB各色のデータ(RGBデータ)と、不透明度 $\alpha$ とを画素毎に出力する。

#### 【0120】

パレット22から出力されたRGBデータは、RGB/YCbCr変換回路29に供給され、各データ長が8ビットの輝度信号Yと色信号Cb、Crのデータに変換される(以下、まとめてYCbCrデータと称する)。これは、以降のプレーン間合成を共通のデータ形式で行う必要があるため、動画データのデータ形式であるYCbCrデータに統一している。

#### 【0121】

RGB/YCbCr変換回路29から出力されたYCbCrデータおよび不透明度データ $\alpha 1$ とがそれぞれ乗算器23に入力される。乗算器23では、入力されたYCbCrデータに不透明度データ $\alpha 1$ が乗ぜられる。乗算結果は、加算器24の一方の入力端に入力される。なお、乗算器23では、YCbCrデータにおける輝度信号Y、色差信号Cb、Crのそれぞれについて、不透明度データ $\alpha 1$ との乗算が行われる。また、不透明度データ $\alpha 1$ の補数( $1 - \alpha 1$ )が乗算器21に供給される。

#### 【0122】

乗算器21では、422/444変換回路20から入力された動画データに不透明度データ $\alpha 1$ の補数( $1 - \alpha 1$ )が乗ぜられる。乗算結果は、加算器24の他方の入力端に入力される。加算器24において、乗算器21および23の乗算結果が加算される。これにより、動画プレーン10と字幕プレーン11とが合成される。加算器24の加算結果が乗算器25に入力される。

#### 【0123】

グラフィクスプレーン12の画像データがRGB/YCbCr変換回路26に入力される。グラフィクスプレーン12の画像データのカラーシステムがRGB(4:4:4)である場合には、カラーシステムがYCbCr(4:4:4)に変換されてRGB/YCbCr変換回路26から出力される。RGB/YCbCr変換回路26から出力されたYCbCrデータが乗算器27に入力される。

#### 【0124】

グラフィクスプレーン12で用いられる画像データがPNG形式である場合には、画像データ中に、画素毎に不透明度データ $\alpha 2$ ( $0 \leq \alpha 2 \leq 1$ )を設定することができる。不透明度データ $\alpha 2$ は、乗算器27に供給される。乗算器27では、RGB/YCbCr変換回路26から入力されたYCbCrデータに対し、輝度信号Y、色差信号Cb、Crのそれぞれについて、不透明度データ $\alpha 2$ との乗算が行われる。乗算器27による乗算結果が加算器28の一方の入力端に入力される。また、不透明度データ $\alpha 2$ の補数( $1 - \alpha 2$ )が乗算器25に供給される。

#### 【0125】

乗算器25では、加算器24の加算結果に対して不透明度データ $\alpha 2$ の補数( $1 - \alpha 2$ )が乗ぜられる。乗算器25の乗算結果は、加算器28の他方の入力端に入力され、上述した乗算器27による乗算結果と加算される。これにより、動画プレーン10と字幕プレーン11との合成結果に対して、さらに、グラフィクスプレーン12が合成される。

#### 【0126】



字幕プレーン 11 およびグラフィクスプレーン 12 において、例えば、表示すべき画像の無い領域の不透明度  $\alpha = 0$  と設定することで、そのプレーンの下に表示されるプレーンを透過表示させることができ、例えば動画プレーン 10 に表示されている動画データを、字幕プレーン 11 やグラフィクスプレーン 12 の背景として表示することができる。

【0127】

なお、この図 17 に示される構成は、ハードウェアおよびソフトウェアの何れでも実現可能なものである。

【0128】

## 2-2. メニュー画面

グラフィクスプレーン 12 には、ユーザに対して操作を促す画面、例えばメニュー画面 10 を表示させることができる。図 20 は、グラフィクスプレーン 12 に表示されるメニュー画面 60 の一例を示す。メニュー画面 60 は、文字や画像を特定の位置に表示させ、ユーザが選択や実行を指示することで新たな動作が開始される「リンク」や「ボタン」を配置することができ、ユーザに対して GUI (Graphical User Interface) を提供する。

【0129】

「リンク」は、例えば、文字列や画像データに対して所定のファイルへのアクセス方法を示す記述を設定し、ユーザがポインティングデバイスなどを用いて画面に表示された当該文字列や画像データ上を指定することで、当該文字列や画像データに設定されたアクセス方法に従って当該ファイルにアクセスできるようにしたものである。「ボタン」は、「リンク」において、例えば通常の状態、選択された状態および押された状態の 3 種類の画像データを用意し、ユーザにより当該ボタン画像が指定された際に、3 種類の画像データの表示を切り換えて、ユーザが行った操作をユーザ自身に認識しやすくしたものである。

【0130】

なお、「リンク」や「ボタン」に対する指定動作は、例えば、マウスを用いて画面上のカーソル表示を移動させ、「リンク」による文字列や画像、「ボタン」による画像上で、マウスボタンを 1 または複数回押すクリック動作を行うことで、なされる。マウス以外のポインティングデバイスによっても、同様の動作を行うことができる。また、これに限らず、リモートコントロールコマンドやキーボードのキー操作により、「リンク」や「ボタン」の指定を行うこともできる。例えば、方向キーなど所定のキーで指定を行いたい「リンク」や「ボタン」を選択し、決定キーなどにより選択された「リンク」や「ボタン」を指定する。

【0131】

図 20 の例では、グラフィクスプレーン 12 上に表示されたメニュー画面 60 の上部に、画像データとしてタイトル 61 が表示される。タイトル 61 に続けて、リンクとしての選択項目 62 A、62 B、62 C および 62 D が表示される。例えばリモートコントロールコマンドのキー操作により、選択項目 62 A、62 B、62 C および 62 D のうち 1 つを選択し指定することで、指定された選択項目にリンクとして設定されたファイルがアクセスされる。

【0132】

メニュー画面 60 の下部に、字幕の表示の有無や、出力される音声を例えば英語および日本語から選択するためのボタン 64 および 65 が表示される。これらボタン 64 や 65 を上述のように操作することで、例えば、それぞれの設定を行う画面を表示するためのファイルがアクセスされ、所定の画面表示がなされる。

【0133】

また、メニュー画面 60 の左下部分には、項目の選択方法を示す文字列 63 が表示される。この文字列 63 も、グラフィクスプレーン 12 上の表示である。

【0134】

この、図 20 に一例が示されるような画面を表示するためには、画面の表示方法やリンク情報などを記述可能な、何らかの記述言語が必要とされる。この発明の実施の第 1 の形態では、この記述言語として、インターネット上の WWW (World Wide Web) など広く普

及している、HTML (Hyper Text Markup Language)を用いる。

【0135】

周知のように、HTML文書では、タグを用いて文書構造が示される。タグは、範囲の開始および終了をそれぞれ示す一対の記号からなり、テキスト中に埋め込んで任意の範囲を指定することができる。例えば、範囲の開始を示す開始タグは、文字列として予め定義された要素を記号「<」と「>」とで囲んで表現され、終了を示す終了タグは、開始を示すタグ中に記される文字列と同一の文字列が記号「</」と「>」とで囲んで表現される。タグ中に、当該タグにより示される要素の属性の記述を含ませることができる。なお、終了タグの記述を省略可能、または、記述しない要素も存在する。

【0136】

HTMLで記述された文書は、一般的には、インターネットに接続されたWebサーバ上に置かれ、Webサーバによりインターネットに対して公開され、インターネットに接続されたパーソナルコンピュータなどのコンピュータ装置に搭載されたブラウザソフトウェアにより閲覧される。この発明の実施の第1の形態では、HTMLの規格に対して独自の拡張を定義して、Blu-ray Discによるメニュー画面表示が可能ようにした。

【0137】

一例として、WWWにおいて、HTML文書中に記述されたリンク情報は、ブラウザソフトウェア上でそのリンクを指定することで、他のHTMLファイルなどにアクセスし読み込むことができる。一方、上述したようなBlu-ray Discのメニュー画面60では、例えばプレイリストの一覧が画像データや文字列、ボタンなどで表示され、あるプレイリストを指定することで、指定されたプレイリストがディスクから読み出され再生されることが期待される。

【0138】

HTMLに対するこれらの拡張は、例えばプログラミング言語において新たな関数を定義することや、新たなAPI (Application Programming Interface)を定義することに相当する。

【0139】

2-3. シナリオについて

上述した図20の例では、メニュー画面60に対してプレイリストの一覧が表示される。このメニュー画面60およびメニュー画面60上の選択項目を選択した際に再生される映像・音声は、実際には、複数のプレイリストが並んだ構造によって構成されている。メニューの一項目を構成する複数のプレイリスト内で、プレイリストの関連付けを行うことで、ストーリーを分岐させる仕組みを実現することができる。ストーリーを分岐できるようにすることで、ユーザの選択によってストーリーが変化していくマルチストーリーや、ユーザの年齢に応じてシーンを差し替えるパレンタル機能を実現することができる。

【0140】

このような機能は、記録済みのディスクにおいて特に有用であって、テレビジョン放送の記録/再生が主目的である現行のBlu-ray Disc規格では想定されていない。

【0141】

なお、画面表示の記述は、HTMLに限らず、例えばHTML 4.0の後継バージョンであるXHTML (extensible HTML)を用いることもできる。これらに限らず、さらに他のマークアップ言語を用いて画面表示の記述を行うこともできる。

【0142】

以下では、この複数のプレイリストが並べられた構造を、シナリオと称する。図21は、シナリオ70の一例の内部構造を示す。このシナリオ70は、複数のプレイリスト73A~73Mを含む。また、シナリオ70は、グラフィクスプレーン12を用いて分岐選択画面を表示する箇所が2箇所(画面80Aおよび80B)設けられている。例えば画面80Aは、分岐選択画面を表示するためのグラフィクスデータ74Aとプレイリスト73C

10

20

30

40

50

を有する。同様に、画面80Bは、分岐選択画面を表示するためのグラフィクスデータ74Bとプレイリスト73Jを有する。

#### 【0143】

このように、シナリオは、プレイリストの並びを決めると共に、グラフィクスプレーン12に対する表示のタイミングを規定する。グラフィクスプレーン12への表示のタイミングは、プレイリスト中のマークを用いて規定することができる。なお、シナリオ70中のイベントハンドラ定義71および72については、後述する。

#### 【0144】

図21の例では、例えば上述のメニュー画面60における選択項目（例えば選択項目62A）は、シナリオ70に一例として「エントリ」と記されたプレイリスト73Aに対応する。すなわち、メニュー画面60において選択項目62Aが指定されると、先ず、シナリオ70のプレイリスト73Aが再生される。プレイリスト73Aの再生が終了されると、続けてプレイリスト73Bが再生され、プレイリスト73Bの再生が終了されると、プレイリスト73Cが再生され、グラフィクスデータ74Aが読み出されて、ユーザにストーリーの分岐選択を促す画面80Aが表示される。

#### 【0145】

画面80Aが表示された以降は、ユーザの選択に基づきストーリーが分岐される。この図21の例では、第1の選択により、画面80Aの後、プレイリスト73D、73E、73Fの順に再生され、全体の再生が終了される。プレイリスト73Fの再生が終了された後、メインのメニュー画面（例えば上述したメニュー画面60）に戻るようにしても良い。

#### 【0146】

画面80Aにおける第2の選択では、画面80Aの後、プレイリスト73Gが再生される。プレイリスト73Gは、例えば所定のタイミングにマークが設定されている。プレイリスト73Gが再生されると、再生装置自身の設定やユーザによる他のシナリオや分岐選択画面などでの選択に基づき、プレイリスト73Gのマーク位置で分岐するか、プレイリスト73Gを最後まで再生するかが制御される。プレイリスト73Gが最後まで再生される場合には、プレイリスト73Gの再生終了後、プレイリスト73M、73Iと再生され、プレイリスト73Jがさらに再生される。

#### 【0147】

一方、プレイリスト73Gにおいてマーク位置で分岐された場合には、次にプレイリスト73K、73Lと再生され、プレイリスト73Lの再生が終了されたら、次に、プレイリスト73Iに設定されたマーク位置から再生が継続される。

#### 【0148】

プレイリスト73Jは、グラフィクスデータ74Bが読み出されて、ユーザにストーリーの分岐選択を促す画面80Bが表示される。画面80Bにおける第1の選択では、上述したプレイリスト73Fが再生される。一方、画面80Bの第2の選択では、プレイリスト73Kに設定されているマーク位置からの再生が行われる。

#### 【0149】

マークの検出、ユーザからの入力、プレーヤの動作変更の検出は、イベントドリブンモデルに基づきなされる。すなわち、プレイリストの再生が開始された、プレイリストの再生が終了された、プレイリストの再生中にマークが検出された、リモートコントロールコマンドのキー操作などによりユーザからの入力があった、などによりイベントが発生される。そして、特定のイベントがきっかけになって実行されるイベントハンドラをプログラム中に用意しておくことで、プレーヤにおいて、当該イベントにより期待される動作が実行される。

#### 【0150】

図21に一例が示されるシナリオ70には、2つのイベントハンドラ71および72が設定されている。これらのうち、イベントハンドラ71は、シナリオ70の全体にわたって有効なイベントハンドラが記述される、グローバルなイベントハンドラ定義である。

## 【0151】

例えば、シナリオ70内のプレイリスト73A~73Mの何れが再生中であっても、リモートコントロールコマンドに設けられたメニューボタンが押されたら、シナリオの一覧が表示されるメニュー画面60が表示される、すなわち、メニュー画面60を表示するためのプレイリストに再生処理が移行される動作を実現したい場合について考える。この場合、リモートコントロールコマンドのメニューボタンが押されたときに発生するイベント(メニューボタン押下イベント)に対応するイベントハンドラとして、メニュー画面60を表示させるプレイリストに処理が移行される命令を、グローバルなイベントハンドラ71として記述する。

## 【0152】

10

一方、イベントハンドラ72は、特定のプレイリストの再生中にのみ、あるいは、特定のユーザ入力画面が表示されている間のみに実行され得るイベントハンドラが記述される、ローカルなイベントハンドラ定義である。例えば、分岐選択画面である画面80Aに表示されているリンクがユーザにより指定されたときに、別のプレイリストを再生するという動作は、リンクが指定されたというイベントに対して、プレイリストの再生を開始する命令をローカルなイベントハンドラ72として記述することで、実現することができる。

## 【0153】

この発明の実施の第1の形態では、上述のようなイベントハンドラ定義を記述するスクリプト言語として、ECMAスクリプトを用いる。ECMAスクリプトは、ECMA(European Computer Manufacturers Association)により定められた、JavaScript(登録商標)に基づいたクロスプラットフォーム用のスクリプト言語である。ECMAスクリプトは、HTML文書との親和性が高いことと、独自のオブジェクトの定義が可能であるため、この発明の実施の第1の形態に用いて好適である。

## 【0154】

2-4. 仮想プレーヤモデルについて

次に、グラフィクスプレーン12が追加され、HTMLとECMAスクリプトとを用いて拡張されたBD/AVフォーマットに基づき動作する再生装置のモデルを考える。以下では、このモデル化された再生装置をBD(Blu-ray disc)仮想プレーヤと称し、このBD仮想プレーヤの構成の定義をBD仮想プレーヤモデルと称する。

30

## 【0155】

図22を用いて、BD仮想プレーヤモデルについて説明する。BD仮想プレーヤ30は、Blu-ray Disc(以下、ディスクと略称する)を再生するもので、パーソナルコンピュータなどのコンピュータ環境上のオブジェクトである。コンピュータ環境は、汎用的なパーソナルコンピュータに限られず、例えば、Blu-ray Discなどの記録媒体を再生するように専用的に構成された再生装置および/または記録再生装置に組み込まれたソフトウェア環境も含む。

## 【0156】

BD仮想プレーヤ30の状態は、概略的には、プレイリストおよびグラフィクスを再生する状態Aと、再生が停止された状態Bの2つの状態に分類できる。ある状態から別の状態への遷移や、ある状態の中での次の動作の指定は、BD仮想プレーヤ30のオブジェクトに対するコマンドにより行われる。

40

## 【0157】

なお、状態Aは、複数の動作を含む。状態Aに含まれる動作としては、高速再生、逆転再生といった変速再生や、ディスク上の任意の時刻から再生する飛び込み再生といった特殊再生などが考えられる。BD仮想プレーヤ30においてグラフィクスプレーン12による表示がなされているときは、これら変速再生や特殊再生が制限されることがある。

## 【0158】

PBC(Play Back Control)プログラム40は、例えばディスクに記録されているシナリオに対応する。詳細は後述するが、シナリオは、ディスクに記録されたプレイリストの

50

再生方法や、メニュー画面の表示を指示する。PBCプログラム40とBD仮想プレーヤ30とは、API(Application Programming Interface)41を介してメソッドのやりとりを行い、ディスクに記録されたプレイリストを再生する。

【0159】

より具体的には、BD仮想プレーヤ30の状態遷移の際には、PBCプログラム40により、API41を介して、BD仮想プレーヤ30において使用目的が定められたメモリとして定義された共通パラメータ32に対して、必要な情報が伝達される。この共通パラメータ32に対する値のセットは、PBCプログラム40とBD仮想プレーヤ30との間でAPI41を介してやりとりされるメソッドである、プレーヤコマンド31により行うことができる。

10

【0160】

この発明の実施の第1の形態では、上述したように、BD仮想プレーヤ30は、イベントドリブンモデルに基づき制御される。BD仮想プレーヤ30の動作中には、様々なイベントが発生する。イベントは、例えば、ユーザによるキー入力やリモートコントロールコマンドによる操作、タイマによる割り込みなどにより、ハードウェア/OS(Operating System)50において発生され、BD仮想プレーヤ30に渡される。これに限らず、再生されたプレイリスト中のマークの検出や、プレーヤの動作変更の検出など、BD仮想プレーヤ30自身によりイベントが発生される場合もある。イベント処理としては、イベントバブルの考え方をを用いる。

【0161】

20

発生されるイベントの種類は、BD仮想プレーヤモデルで定義されている。イベントが発生すると、スクリプトプログラムとしてイベントハンドラが記述されていれば、発生されたイベントに対応するイベントハンドラが実行される。プログラムとしてイベントハンドラが記述されていない場合には、デフォルトのイベントハンドラが実行され、規格で規定されているプレーヤ動作が実行される。

【0162】

プレーヤ動作に関するメソッドが実行された場合、プレーヤ動作が正常に開始されたか否かは、メソッドに対する戻り値で知ることができる。一方、指示された命令が何時終了されたかは、イベント発生により知るしかない。そのため、例えばシナリオ終了の再生を開始するメソッドをスクリプトプログラムに記述した場合には、シナリオ終了のイベントを受け取って実行されるイベントハンドラを共に記述することがより望ましい。

30

【0163】

なお、同一のイベントであっても、シナリオ毎に異なるイベントハンドラを記述することができる。

【0164】

BD仮想プレーヤ30における割り込みイベントは、(1)再生中のコンテンツから発せられるイベント、(2)ユーザからの割り込みにより発生されるイベント、(3)プレーヤの状態の変化により発生されるイベント、の3種類に大別される。

【0165】

(1)の、再生中のコンテンツから発せられるイベントは、予め設定された割り込みであって、当該コンテンツの再生の度に、同一のタイミングで発生される。例えば、BD仮想プレーヤ30においてプレイリストを再生中に、プレイリスト中に記述されたマークで指し示したディスク上の時刻に到達すると、BD仮想プレーヤ30においてマーク検出の割り込みが発生される。また、スクリプトにおいてタイマが設定された場合に、設定された時刻、例えば当該スクリプトによりタイマ設定がなされた10秒後にタイマ割り込みのイベントが発生される。

40

【0166】

(2)の、ユーザからの割り込みによるイベントは、発生および発生タイミングの予測ができないイベントである。例えば、ユーザによるリモートコントロールコマンドのキー操作により、この割り込みイベントが発生される。この場合、ユーザが何時キー操作を行

50

うかが分からないため、事前にそのタイミングを知ることができない。

【0167】

(3) の、BD仮想プレーヤ30の状態変化によるイベントは、例えば、音声や字幕のストリーム変更を通知するためのイベントである。コンテンツの再生状態から停止状態へ、停止状態から再生状態への状態遷移でも、このイベントが発生される。このプレーヤの状態変化によるイベントは、上述した(1)の再生中のコンテンツから発せられるイベントや、(2)のユーザによる割り込みイベントに伴い引き起こされる場合も多い。上述の音声や字幕のストリーム変更の通知のイベントの例では、ユーザによるリモートコントロールコマンドなどのキー操作による割り込みイベントで、音声や字幕のストリームが変更され、この変更に伴うBD仮想プレーヤ30の状態遷移に基づき、変更通知のイベントが発生される。 10

【0168】

この発明の実施の第1の形態では、グラフィクスプレーン12を用いた表示制御がHTML(Hyper Text Markup Language)4.01またはXHTML(eXtensible HTML)文書として記述され、グラフィクスプレーン12による表示画面に対するイベントは、HTML4.0の組み込みイベントが用いられる。また、HTML4.0の組み込みイベントでは不十分なイベントは、ECMASクリプトを用いて記述される。

【0169】

HTML形式とECMASクリプトを組み合わせて用いる場合、イベントが発生すると、まず、そのイベントが発生した要素に属性で指定されているイベントハンドラが文書中に存在するか否かで、処理が異なる。イベントハンドラが存在する場合は、そのイベントハンドラが実行される。イベントハンドラが存在しない場合は、文書中にグローバルなイベントハンドラが用意されているか否かが探される。その結果、若し、グローバルなイベントハンドラが当該文書中に存在すれば、そのイベントハンドラが実行される。一方、文書中にスクリプト言語によるイベントハンドラの記述が無い場合には、BD仮想プレーヤ30において当該イベントに対して用意されたデフォルトのイベント処理が行われる。 20

【0170】

イベントハンドラの記述方法としては、HTML文書における要素の属性として記述する方法と、ECMASクリプトにおいてメソッドcaptureEventsを使って記述する方法とを用いることができる。 30

【0171】

HTML文書を用いてイベントハンドラを記述する方法について説明する。この発明の実施の第1の形態によるBD仮想プレーヤ30では、HTML4.0に規定される組み込みイベントのうち、例えばイベントonload、イベントonunload、イベントonclickおよびイベントonkeypressを用いることができる。これらのイベントは、タグの要素中に属性として記述される。

【0172】

イベントonloadは、利用者エージェントが1つのウィンドウまたは1対のタグ<FRAMESET></FRAMESET>で定義される全てのフレームを終了する際に発生される。例えば、メニュー画面の表示の際にイベントonloadが発生される。 40

【0173】

なお、ウィンドウは、HTMLの規定に基づきHTMLファイルを表示するブラウザアプリケーションにおける表示単位であり、フレームは、1つのウィンドウを分割して複数のHTMLファイルを表示させる際に用いられる枠組みである。ここでは、フレーム内に表示されるHTMLファイルも含めてフレームと称する。このイベントonload属性は、要素BODYおよび要素FRAMESETと共に使用することができる。

【0174】

イベントonunloadは、利用者エージェントが1つのウィンドウまたは1つのフレームから1つのHTML文書を除去する際に発生される。このイベントonunload属性は、要素BODYおよび要素FRAMESETと共に使用することができる。 50

## 【0175】

イベントonclickは、ポインティングデバイスなどにより要素が指定された場合に発生される。例えば、マウスボタンのクリック操作を行った場合に、イベントonclickが発生される。このイベントonclick属性は、HTML 4. 0における殆どの要素と共に用いることができる。

## 【0176】

イベントonkeypressは、キーが要素上で押されたり解放されたりする際に発生される。例えば、ある要素により定義される画面上の領域が選択状態のときに、キーボード上の所定のキーやリモートコントロールコマンドのキーが押された際に、イベントonkeypressが発生される。このイベントonkeypress属性は、HTML 4. 0における殆どの要素と共に用いることができる。

## 【0177】

BD仮想プレーヤ30の動作を制御するためには、上述のHTMLによるイベントでは不十分であるため、この発明の実施の第1の形態では、独自のイベントが定義される。図23は、BD仮想プレーヤ30において独自に定義される一例のイベントを示す。このイベントは、ECMAスクリプトにより、HTML文書に埋め込まれて記述される。また、イベントハンドラを指定する属性名は、イベント名の先頭に「on」を付して示す。

## 【0178】

イベントTimerFiredは、カウントダウンタイマの値が「0」になるか、カウントアップタイマの値が指定された値になったときに発生される。イベントPlayStoppedおよびイベントPlayStilledは、それぞれ再生の停止および一時停止により発生される。イベントStillReleasedは、一時停止が解除された際に発生される。イベントPlayPausedおよびイベントPauseReleasedは、それぞれユーザによる再生の一時停止および一時停止の解除の際に発生される。イベントPlayRepeatedは、繰り返し再生時の先頭が検出された際に発生される。

## 【0179】

イベントSPDisplayStatusChangedは、サブピクチャ（字幕）ストリームの表示／非表示状態が変化した際に発生される。イベントSelectedAudioChangedおよびイベントVideoStoppedは、それぞれ再生するオーディオストリームおよびビデオストリームが変わった際に発生される。

## 【0180】

イベントScenarioStartedおよびイベントScenarioEndedは、それぞれシナリオの先頭および終端が検出された際に発生される。イベントPlaylistStartedおよびイベントPlaylistEndedは、それぞれプレイリストの先頭および終端が検出された際に発生される。イベントPlayItemStartedおよびイベントPlayItemEndedは、それぞれプレイアイテムの先頭および終端が検出された際に発生される。

## 【0181】

イベントMarkEncounteredは、プレイリストの再生中にマークが検出された際に発生される。このイベントは、例えばグラフィクスプレーン12に画像データを表示する際に用いられる。共通パラメータ32に、検出されたマークの種類と番号とが格納される。

## 【0182】

イベントButtonPressedは、画面上に配置されたボタンが押された際に発生される。例えば、グラフィクスプレーン12上に表示されたボタンを、キー操作やマウスのクリック操作などにより仮想的に押した場合に、イベントButtonPressedが発生される。

## 【0183】

イベントValidPeriodStartedは、有効期間が開始された際に発生される。これは、リンクの選択有効期間を設定する際に用いることができる。また、イベントValidPeriodEndedは、有効期間が終了した際に発生される。これは、リンクの強制実行のときに用いることができる。

## 【0184】



イベントKeyPressedは、リモートコントロールコマンドのキーが押されたときに発生される。押されたキーの種類は、イベントハンドラ内の「switch」文などで識別される。

【0185】

2-5. メソッドについて

図24、図25および図26は、この発明の実施の第1の形態で定義される、BD仮想プレーヤ30が有するメソッドの一例を示す。これらのメソッドにより、BD仮想プレーヤ30における動作および状態、BD仮想プレーヤ30で扱うビデオストリーム、オーディオストリームおよびサブピクチャ（字幕プレーン11の画像データ）に関する情報取得や制御、共通パラメータ32に対する操作、タイマやキー入力の割り込みに応じた処理、グラフィクスプレーン12で扱う画像データの制御がそれぞれ定義される。

10

【0186】

これらのメソッドをPBCプログラム中で適宜用いることにより、BD仮想プレーヤ30の動作を制御することが可能とされる。これらのメソッドは、図22におけるAPI41に実装される。PBCプログラムの記述に基づきこれらのメソッドがAPI41から呼び出され、BD仮想プレーヤ30におけるディスク再生が制御される。PBCプログラムの具体的な例については、後述する。

【0187】

プレーヤ動作に関するメソッドについて説明する。メソッドplayScenario(scenarioNumber,[scenarioTime])で、"scenarioNumber"で指定されるシナリオの再生が指示される。"scenarioNumber"は、実際には、シナリオ構成を記述したファイルなどの場所を示すURI (Universal Resource Identifier)になる。メソッドplayPlaylist(playlistNumber)で、"playlistNumber"で指定されたプレイリストの再生が指示される。メソッドplayChapterMark(playlistNumber,chapterNumber)で、"playlistNumber"で示されるプレイリスト中の、"chapterNumber"指定されるチャプタからの再生が指示される。メソッドplayPlayItem(playlistNumber,playItemNumber)で、"playlistNumber"および"playItemNumber"で示されるプレイアイテムからの再生が指示される。"playItemNumber"は、"playItem\_id"であり、値「0」を指定することで、当該プレイアイテムが属するプレイリストの先頭から再生される。

20

【0188】

メソッドplay(position)(object)で、シナリオ内での移動が指示される。すなわち、現在移動中の箇所から前後のプレイリストやプレイアイテムに移動することがこのメソッドで指示される。なお、パラメータ"position"は、"prev"、"next"、"top"、"goUp"または"tail"の何れかの値を取り、パラメータ"object"で示される移動対象（プレイリスト、プレイアイテムまたはチャプタ）に対する移動方法が指示される。

30

【0189】

メソッドstop()で、シナリオ再生の停止が指示される。この場合、標準レジスタの値は、保持されない。メソッドresume()で、前回再生を停止した箇所からの再生の再開が指示される。メソッドplaySoundEffect()で、選択された効果音の再生が指示される。

【0190】

プレーヤ状態に関するメソッドについて説明する。図25において、メソッド getMenuDescriptionLanguage()で、メニュー表示の際に用いられる言語が取得される。メソッド getScenarioNumber()、メソッド getPlaylistNumber()およびメソッド getChapterNumber()で、再生中のシナリオ番号、プレイリスト番号およびチャプタ番号がそれぞれ取得される。メソッド getPlayerSupport()で、プレーヤのバージョン情報が取得される。

40

【0191】

ビデオストリームに関するメソッドについて説明する。メソッド setVideoStreamNumber()で、デコードするビデオストリームが指定される。メソッド getVideoStreamNumber()、メソッド getVideoStreamStatus()およびメソッド getVideoStreamAttr()で、再生中のビデオストリームの番号、状態および属性がそれぞれ取得される。なお、ビデオストリームの属性は、例えば、そのビデオストリームの符号化方式、解像度、アスペクト比、アスペク

50



ト比が4 : 3のときのディスプレイモード、クローズドキャプションの有無である。メソッドsetAngleNumber()で、アングル番号が指定される。メソッドgetAngleNumber()で、選択中のアングル番号が取得される。メソッドgetMaxVideoStream()で、ビデオストリームの最大数が取得される。

#### 【0192】

オーディオストリームに関するメソッドについて説明する。メソッドgetAudioStreamAvailability()で、指定のオーディオストリームが含まれているか否かが取得される。メソッドgetAudioStreamLanguage()で、指定のオーディオストリームの言語に関する情報が取得される。メソッドgetAudioStreamStatus()で、指定のオーディオストリームの状態が取得され、メソッドsetAudioStreamStatus()で、指定のオーディオストリームに対する状態のセットが指示される。なお、オーディオストリームの状態は、例えば再生または非再生である。メソッドgetAudioStreamAttribute()で、指定のオーディオストリームの属性が取得される。

#### 【0193】

サブピクチャストリーム（字幕データ）ストリームに関するメソッドについて説明する。メソッドgetSPStreamAvailability()で、指定のサブピクチャストリームが含まれているか否かが取得される。メソッドgetSPStreamLanguage()で、指定のサブピクチャストリームで用いられている言語が取得される。メソッドgetSPDisplayStatus()で、サブピクチャストリームの表示状態が取得され、メソッドsetSPDisplayStatus()で、サブピクチャストリームの表示状態がセットされる。サブピクチャストリームの表示状態は、例えばサブピクチャストリームの表示のオン／オフである。メソッドgetSpStreamAttribute()で、指定のサブピクチャストリームの属性が取得される。サブピクチャストリームの属性は、例えばサブピクチャストリームの表示がアスペクト比4 : 3またはワイド画面の何れでなされるかである。

#### 【0194】

共通パラメータ32に関するメソッドについて説明する。図26では、レジスタリード／ライトとして記されている。メソッドclearReg()で、BD仮想プレーヤ30が内蔵しているメモリ領域すなわちレジスタについて、全レジスタの初期化が指示される。メソッドsetReg()で、指定のレジスタへの値のセットが指示される。メソッドgetReg()で、指定のレジスタからの値の読み出しが指示される。

#### 【0195】

タイマに関するメソッドについて説明する。メソッドsleep()で、指定された時間の処理の停止が指示される。メソッドsetTimeout()で、指定された時間後に関数や処理を実行することが指示される。メソッドsetInterval()で、指定された時間毎に処理を実行することが指示される。なお、タイマに関するメソッドにおいて、時間はミリ秒単位で指定することができる。メソッドclearTimer()で、指定された登録タイマIDの処理の中止が指示される。メソッドpauseTimer()で、指定された登録タイマIDのタイマの一時停止が指示される。メソッドresumeTimer()で、指定された登録タイマIDのタイマを一時停止から再開することが指示される。

#### 【0196】

キー入力に関するメソッドでは、メソッドgetPressedKey()で、入力された（押された）キーの種類が取得される。

#### 【0197】

グラフィクスに関するメソッドについて説明する。メソッドloadGraphics(htmlfile, ID)で、"htmlfile"で示されるファイルを読み込み、グラフィクスプレーン12に非表示で展開することが指示される。展開されたグラフィクス画像には"ID"が割り当てられ、後述のメソッドで参照される。メソッドshowGraphics(ID)は、上述のメソッドloadGraphics(htmlfile, ID)により既にグラフィクスプレーン12上に展開されている画像に対して、表示が指示される。メソッドhideGraphics(ID)で、"ID"で示される画像の表示を消去することが指示される。

## 【0198】

その他のメソッドについて説明する。メソッドrandom(input Number num)で、1から"num"までの乱数の発生が指示される。乱数の発生は、独自の定義で行われる。メソッドcatchEvent(eventname, eventhandler)で、"eventname"で示されたイベントが発生した際、"eventhandler"で示された関数を実行することが指示される。

## 【0199】

## 2-6. シナリオの実行について

次に、PBCプログラム40について、より具体的に説明する。この発明の実施の第1の形態では、一つのシナリオに対して、一つのスクリプトファイルが作成される。また、グラフィクスプレーン12を用いてメニュー画面60などを表示する場合には、1画面に 10  
 に対して一つのHTMLファイルが作成される。スクリプトファイルおよびHTMLファイルは、ファイル名の拡張子がそれぞれ「js」および「html」とされ、拡張子でこれら2種類のファイルを識別可能とする。なお、ディスクがドライブ装置に挿入されて最初に実行されるスクリプトプログラムが格納されるファイルは、ファイル名を例えば「startup.js」に固定的とする。

## 【0200】

一例として、図27に示されるようなシナリオ構成のディスクを考える。このディスクは、シナリオScenario000およびScenario001の2つのシナリオを有し、シナリオScenario000は、シナリオScenario001へのリンク 20  
 ボタン92を表示するメニュー画面91を表示させる。ディスクが再生装置に装填されると、メニュー画面91が表示され、メニュー画面91に表示されるボタン92が例えばクリック操作により押されると、シナリオScenario001が再生される。シナリオScenario001の再生が終了されると、再びメニュー画面91が表示される。

## 【0201】

図28は、図27の構成に対して必要とされる一例のファイルを一覧して示す。この例では、ファイル「startup.js」、ファイル「scenario000.js」、ファイル「000.html」、ファイル「00000.rpls」、ファイル「scenario001.js」およびファイル「00001.rpls」の 6ファイルが必要とされる。

## 【0202】

これらのうち、ファイル「scenario000.js」は、シナリオScenario000の構成 30  
 情報が記述されるスクリプトファイルであって、メニュー画面91すなわちシナリオ一覧画面の構成情報が記述される。ファイル「000.html」は、メニュー画面91のレイアウト情報が記述されるHTMLファイルである。ファイル「00000.rpls」は、メニュー画面91において背景として表示されるプレイリストファイルである。ファイル「scenario001.js」は、シナリオScenario001の構成情報が記述されるスクリプトファイルである。ファイル「00001.rpls」は、シナリオScenario001で再生されるプレイリストの情報が記述されたプレイリストファイルである。

## 【0203】

なお、図28では、プレイリストファイル「00000.rpls」および「000001.rpls」で実際に再生されるコンテンツファイルなど（クリップ情報ファイル「%%%%.clip」、AVストリームファイル「\*\*\*\*\*.m2ts」）については省略されている。 40

## 【0204】

図28に示される各ファイルは、図29に一例が示されるディレクトリ構造に従い、ディスクに記録される。

## 【0205】

図30～図33は、スクリプトファイルおよびHTMLファイルのより具体的な記述の例を示す。図30は、ファイル「startup.js」の記述例である。ファイル「startup.js」では、当該ディスクに格納されているシナリオの数と名前とが定義される。そして、メソッドplayScenario("scenario000")の記述により、ファイル「scenario000.js」が呼び出され、シナリオScenario000の再生が指示される。シナリオScenario 50

000が再生されることにより、メニュー画面91が表示される。

#### 【0206】

図31は、ファイル「scenario000.js」の記述例である。関数functionが定義され、メソッドcatchEvent()に「on」が付されて記述されたイベント（図23参照）が発生すると、対応する関数functionで定義された内容が実行される。図31の例では、メニュー画面91のレイアウト情報が記述されたHTMLファイル「000.html」が指定され、グラフィクスプレーン12へのメニュー画面91の読み込みおよび表示のタイミングが制御される。それと共に、メソッドplayPlayList("00000.rpls")が記述され、プレイリスト「00000.rpls」の再生が指示される。

#### 【0207】

すなわち、このファイル「scenario000.js」により、プレイリスト「00000.rpls」による動画の動画プレーン10に対する表示が指示されると共に、プレイリスト「00000.rpls」の再生中に検出されたマークのタイミングで、メニュー画面91をグラフィクスプレーン12に表示する旨が指示される。

#### 【0208】

図32は、ファイル「000.html」の記述例である。タグ<style type="text/css">および</style>で囲まれた部分に、「menu」および「scenario001」で参照される画像の、メニュー画面91におけるレイアウト情報が記述される。この図32の例では、レイアウト情報が画面上の絶対座標で記述され、画像名「menu」で参照される画像データが画面の上端から200ピクセル、左端から800ピクセルの位置に、幅200ピクセル、高さ50ピクセルの画像として表示されることが示される。同様に、画像名「scenario」で参照される画像データが画面の上端から700ピクセル、左端から700ピクセルの位置に、幅400ピクセル、高さ100ピクセルの画像として表示されることが示される。

#### 【0209】

タグ<script type="text/javascript">および</script>で囲まれた部分に、関数functionで、マウス操作に対するイベントハンドラonMoverhandler(f)、onMouthandler(f)およびonMclickhandler(f)がそれぞれ定義される。この図32の例では、イベントハンドラonMoverhandler(f)、onMouthandler(f)およびonMclickhandler(f)に対してボタン画像を表示する画像データである画像データ「201.png」、「200.png」および「202.png」がそれぞれ対応付けられると共に、イベントハンドラonMclickhandler(f)においては、シナリオファイル「scenario001.js」の再生が指示される。

#### 【0210】

タグ<body>および</body>で囲まれた部分に、メニュー画面91においてグラフィクスプレーン12に表示される画像データが指定される。上述のタグ<style type="text/css">および</style>で囲まれた部分に記述される画像名に対応する画像データのファイル名（「100.png」、「200.png」）がそれぞれ記述される。さらに、画像名「scenario001」で参照される画像データに対しては、マウスなどポインティングデバイスに対する操作に応じて発生するイベントonMouseover、onMouseoutおよびonclickに対して、上述の関数functionで定義されたイベントハンドラonMoverhandler(f)、onMouthandler(f)およびonMclickhandler(f)がそれぞれ実行されることが指示される。

#### 【0211】

なお、イベントonMouseoverは、指定された領域などにカーソル表示が重なった際に、イベントonMouseoutは、指定された領域などからカーソル表示が離れた際に、また、イベントonclickは、指定された領域にカーソル表示があるときにポインティングデバイスに対する所定の操作、例えばマウスボタンのクリックが行われた際に、それぞれ発生されるイベントである。

#### 【0212】

図33は、ファイル「scenario001.js」の記述例である。関数function UOPControl()において、このシナリオファイル「scenario001.js」の再生中に、例えばリモートコント

10

20

30

40

50

ロールコマンドに設けられたメニューキーが押されたときに、上述した、メニュー画面 9 1 を表示するためのシナリオ `Scenario000` を再生することが定義される。また、関数 `function playListEnded()` により、このシナリオファイル「`scenario001.js`」によるプレイリストの再生が終了したら、メニュー画面 9 1 を表示するためのシナリオ `Scenario000` を再生することが定義される。また、このシナリオファイル「`scenario001.js`」でプレイリスト「`00001.rpls`」が再生されることが指示される。

#### 【0213】

上述した図 30～図 33 の記述に基づく動作について、概略的に説明する。ディスクが再生装置に装填されると、まず、ファイル「`startup.js`」が読み込まれ、このファイル中の記述によりファイル「`scenario000.js`」が呼び出される。このファイルに記述されたシナリオ「`scenario000`」が実行されると、図 31 の記述に基づき、プレイリスト「`00000.rpls`」による動画が動画プレーン 10 に表示される。

#### 【0214】

さらに、プレイリスト「`00000.rpls`」中に指定されたマークに対応するタイミングで、ファイル「`000.html`」が呼び出され、ファイル「`000.html`」の記述に基づきシナリオ一覧を表示するメニュー画面 9 1 がグラフィクスプレーン 12 上に展開され、表示される。このメニュー画面 9 1 自体も、シナリオ「`scenario000`」という一つのシナリオで構成されたものである。

#### 【0215】

メニュー画面 9 1 には、例えば「Menu」という文字列が描画された画像ファイル「`100.png`」と、「Scenario001」という文字列が描画された画像ファイル「`200.png`」とが配置されている。これらの画像ファイルは、グラフィクスプレーン 12 上に配置され表示される。グラフィクスプレーン 12 の背景に表示される動画プレーン 10 上では、プレイリスト「`00000.rpls`」による動画が表示される。動画プレーン 10 上のプレイリスト「`00000.rpls`」による動画と、グラフィクスプレーン 12 上のファイル「`000.html`」によるメニュー画面 9 1 とが重ね合わされて同一画面上に表示される。すなわち、動画を背景にメニュー画面 9 1 が表示される。

#### 【0216】

このとき、グラフィクスプレーン 12 上に表示されている画面（メニュー画面 9 1）に所定に不透明度を設定し、メニュー画面 9 1 を動画プレーン 10 上の動画に対して透過させて表示させることができる。なお、この例では、例えばプレイリスト「`00000.rpls`」の先頭および末尾にマークが設定され、メニュー画面 9 1 は、プレイリスト「`00000.rpls`」の再生と共に表示が開始され、プレイリスト「`00000.rpls`」の再生終了と共に表示が消去されるようになっている。

#### 【0217】

メニュー画面 9 1 上には、ユーザの例えばリモートコントロールコマンドのキー操作に応じて移動可能なカーソルが表示される。上述の画像ファイル「`200.png`」の表示にカーソル表示が重なると、ファイル「`000.html`」において定義されるイベント `Mouseover` が発生される。このイベント `Mouseover` が発生されると、当該画像ファイル「`200.png`」にフォーカスが当たっていることを表現するために、イベント `onMouseover` に対応するイベントハンドラ `onMoverhandler()` が実行される。この、イベントハンドラ `onMoverhandler()` の実行により、画像ファイル「`200.png`」が画像ファイル「`201.png`」に置き換えられる。画像ファイル「`201.png`」は、例えば画像ファイル「`200.png`」に対して配色が変えられたボタン画像である。

#### 【0218】

カーソル表示が画像ファイル「`201.png`」上にあるときに、ユーザにより、リモートコントロールコマンドの所定のキー操作によりクリック操作が行われると、イベント `onclick` に対応するイベントハンドラ `onMclickehandler()` が実行され、画像ファイル「`201.png`」が、選択されたことを表現する画像ファイル「`202.png`」に置き換えられる。画像ファイル「`202.png`」は、例えばボタンが押されたことを仮想的に表現するようなボタン画像で

ある。

【0219】

このように、「フォーカスされた」、「クリックされた」というイベント発生に対応するイベントハンドラをファイル「000.html」中に記述することで、ユーザの入力に反応するインタラクティブな機能を有するメニュー画面が実現される。

【0220】

上述のようにして、メニュー画面91において、「Scenario001」と描画されたボタン画像がクリックされると、シナリオ「Scenario001」の再生処理に移行される。ファイル「scenario001.js」が実行されることで、シナリオ「Scenario001」が再生される。図33に示したように、ファイル「scenario001.js」中に記述されたメソッドplayPlaylist("00001.rpls")が呼び出され、プレイリスト「00001.rpls」が再生される。 10

【0221】

プレイリスト「00001.rpls」の再生が終了されると、プレイリスト再生終了のイベントPlayListEnded()が発生され、このイベントに対応したイベントハンドラplayScenario("scenario000.js")により、シナリオ「Scenario000.js」を再生するように指示される。すなわち、この例では、シナリオ「Scenario001」の再生が終了されると、再びメニュー画面91が表示される。

【0222】

なお、シナリオ「Scenario001」の再生中に、「keyID」で指定されるキーが操作されても、シナリオ「Scenario000.js」の再生が指示され、再びメニュー画面91が表示される。 20

【0223】

なお、図30～図33で示したHTMLおよびECMAスクリプトの記述は、一例であって、これはこの例に限定されるものではない。すなわち、HTMLやECMAスクリプトの記述には自由度があり、一部を異なる記述にしても同等の動作を実現するようにできる。

【0224】

2-7. デコーダモデル

図34は、この発明の実施の第1の形態に適用できるプレーヤデコーダ100の一例の構成を示す機能ブロック図である。このプレーヤデコーダ100は、図示されないドライブ装置に装填されたディスクから再生されたデータを解釈し、AVストリームを出力すると共に、出力されたAVストリームに対するユーザによるインタラクティブな操作を可能とする。 30

【0225】

なお、プレーヤデコーダ100は、図示されないCPUにより全体の動作が制御される。例えば、プレーヤデコーダ100の各部におけるストリームやデータの流れは、CPUにより監視され、制御される。

【0226】

図示されないドライブ装置にディスクが装填されると、上述したように、まずファイル「startup.js」が再生され、このファイル「startup.js」の記述に基づき、シナリオファイルが読み出され、さらに、シナリオファイルの記述に従い、HTMLファイル、動画プレーン10に表示するための動画データ、字幕プレーン11やグラフィクスプレーン12に表示するための画像データ、プレイリストファイル、シナリオファイルから呼び出される別のシナリオファイル、音声データなどがディスクから読み出される。 40

【0227】

以下では、ディスクから読み出されるこれらのデータのうち、動画データ、サブピクチャ（字幕データ）や音声データといった、連続的に処理する必要があるストリームをリアルタイムストリームと称する。また、シナリオファイル、HTMLファイル、画像データファイル、プレイリストファイルといった、連続的な処理を要求されない非リアルタイムな 50

データを、ストアオブジェクトと称する。ストアオブジェクトは、メモリ上などに蓄積、展開され、必要に応じて処理される。

#### 【0228】

プレーヤデコーダ100は、チャンネル(1)および(2)の2系統の入力チャンネルを有し、入力チャンネル(1)の入力端101に、ストアオブジェクトが入力される。入力チャンネル(2)の入力端202に、リアルタイムストリームが入力される。入力端202に、ストアオブジェクトを入力することも可能である。この実施の第1の形態では、入力端202に入力されるリアルタイムストリームおよび一部のストアオブジェクトは、例えばMPEG2 TSである。

#### 【0229】

なお、入力端202に入力されるリアルタイムストリームは、MPEG2 TSに限られない。パケット単位で伝送され、ビデオデータ、オーディオデータ、静止画像データなどを多重化可能であれば、他の形式のストリームを入力するようにしてもよい。このときには、後述するPIDフィルタ110は、そのストリーム形式に適合したデマルチプレクサとして用いられ、ビデオデータ、オーディオデータ、静止画像データなどを分離する。

#### 【0230】

また、例えば、ドライブ装置においてディスクの回転速度を2倍速などの高速回転としてディスクからの読み出し転送レートを上げ、時分割で動作させることにより、ディスクからの、チャンネル(1)および(2)の2系統の読み出しが実現可能である。

#### 【0231】

先ず、入力チャンネル(1)の系統について説明する。入力端101に入力されたストアオブジェクトは、スイッチ回路102に入力される。ストアオブジェクトとしてECMAスクリプトやHTMLファイルなどによるプログラムコードが入力された場合、スイッチ回路102において出力端102Aが選択され、入力されたプログラムコードがコードバッファ104に蓄えられる。

#### 【0232】

一方、ストアオブジェクトとして画像データが入力された場合、スイッチ回路102において出力端102Bが選択され、入力された画像データがスイッチ回路103に入力される。入力端202に入力されたリアルタイムストリームに、字幕プレーン11やグラフィックスプレーン12に表示するための画像データが含まれていない場合には、スイッチ回路103で入力端103Aが選択され、スイッチ回路102から入力された画像データがコンテンツバッファ105に蓄えられる。

#### 【0233】

同様に、入力端202に入力されたリアルタイムストリームに、字幕プレーン11やグラフィックスプレーン12に表示するための画像データが含まれている場合には、スイッチ回路103において入力端103Bが選択され、当該画像データがコンテンツバッファ105に蓄えられる。コードバッファ104およびコンテンツバッファ105に蓄えられたストアオブジェクトは、必要に応じて読み出され、マルチメディアエンジン106に供給される。

#### 【0234】

コンテンツバッファ105に蓄えられたストアオブジェクトのうち画像データは、スイッチ回路107および108をそれぞれ介して、サブピクチャデコーダ116および静止画像デコーダ117にも供給される。

#### 【0235】

マルチメディアエンジン106は、XMLパーサ106A、スクリプトインタプリタ106Bおよびグラフィックスレンダラ106Cを含む。マルチメディアエンジン106は、独立的なハードウェアで構成してもよいし、上述した図示されないCPUの、所定のプログラムに基づく処理で実現することも可能である。

#### 【0236】

XMLパーサ106Aは、XML(Extensible Markup Language)文書を解析する機能を

10

20

30

40

50

有し、HTML文書の解析も可能である。XMLパーサ106Aで解釈されたHTML文書は、このプレーヤデコーダ100で実行可能な形式に変換される。スクリプトインタプリタ106Bは、ECMAスクリプトを解析し、このプレーヤデコーダ100で実行可能な形式に変換される。また、グラフィクスレンダラ106Cは、画像データを、字幕プレーン11およびグラフィクスプレーン12に展開可能な形式にデコードする。

#### 【0237】

マルチメディアエンジン106において、バッファ109をワークメモリとして、これらXMLパーサ106A、スクリプトインタプリタ106Bおよびグラフィクスレンダラ106Cの処理が行われる。例えば、XMLパーサ106Aおよびスクリプトインタプリタ106Bにより、バッファ109のうちコードバッファ109Aが用いられる。また、10  
グラフィクスレンダラ106Cにより、バッファ109のうちグラフィクスバッファ109Dが用いられる。バッファ109は、上述のコードバッファ109Aおよびグラフィクスバッファ109Dの他に、文字列の表示に用いるフォントデータが格納されるフォントバッファ109B、XMLパーサ106AでHTML文書を解析した結果を階層化された木構造で保持するためのツリーバッファ109Cなどが含まれる。

#### 【0238】

マルチメディアエンジン106では、例えば、コードバッファ104に蓄えられたECMAスクリプトを読み出し、読み出されたECMAスクリプトの記述に基づき、必要に応じて、コードバッファ104からの他のECMAスクリプトやHTML文書の読み出し、  
20  
コンテンツバッファ105からの画像データの読み出しなどを行う。コードバッファ104およびコンテンツバッファ105に格納されたデータは、当該データが不要になるまで、コードバッファ104やコンテンツバッファ105に保持しておくことができる。したがって、これらコードバッファ104やコンテンツバッファ105に格納されたデータは、必要に応じて何度でも読み出して使うことができる。

#### 【0239】

マルチメディアエンジン106では、上述の他にも、入力された複数種類のデータのデマルチプレクス処理、JavaVM（Java（登録商標）仮想マシン）機能などが行われる。さらに、マルチメディアエンジン106により、ユーザからの、リモートコントロールコマンドやポインティングデバイスなどによる入力を受け取られ、所定に処理される。ユーザ入力は、さらに、後述するサブピクチャデコーダ116、静止画像デコーダ11  
30  
7、オーディオデコーダ118、MPEGビデオデコーダ120およびシステムデコーダ121にも供給される。

#### 【0240】

グラフィクスレンダラ106Cで処理された画像データは、スイッチ回路130および131をそれぞれ介してサブピクチャプレーン132および静止画像プレーン134に供給されると共に、グラフィクスプレーン133およびM/Sスイッチプレーン136に供給される。なお、この例では、サブピクチャプレーン132およびグラフィクスプレーン133に供給される画像データは、PNG形式であるものとする。これらの各プレーン132、133、134および136に画像データが供給されるタイミングは、マルチメディアエンジン106により制御される。なお、M/Sスイッチプレーン136は、スイッチ回路137を制御し、後述する静止画像プレーン134から出力される静止画像データ  
40  
と、動画像プレーン135から出力される動画像データとを切り換えて後段に供給する。

#### 【0241】

ここで、サブピクチャプレーン132およびグラフィクスプレーン133は、それぞれ上述した字幕プレーン11およびグラフィクスプレーン12に対応する。静止画像プレーン134および/または動画像プレーン135は、上述した動画像プレーン10に対応する。なお、サブピクチャプレーン132、グラフィクスプレーン133、静止画像プレーン134および動画像プレーン135は、例えばフレームメモリからなる。

#### 【0242】

マルチメディアエンジン106は、さらに、後述するプレゼンテーションプロセッサ1  
50



39に対して、静止画像プレーン134または動画像プレーン135、サブピクチャプレーン132、ならびに、グラフィクスプレーン133を切り換える制御信号を供給する。同様に、マルチメディアエンジン106は、後述するプレゼンテーションプロセッサ141に対して、オーディオストリーム出力を制御するような制御信号を供給する。

#### 【0243】

次に、入力チャンネル(2)の系統について説明する。入力端202にMPEG2 TSで入力されたりアルタイムストリームは、PIDフィルタ110に供給され、MPEG2 TSのトランスポートパケットに格納されるPID(Packet Identification)が抽出され、当該トランスポートパケットに格納されるストリームの属性が検出される。PIDフィルタ110では、このストリーム属性に基づき、入力されたりアルタイムストリームが、トランスポートパケット毎に対応する系統に振り分けられる。 10

#### 【0244】

PIDに基づき、トランスポートパケットがストアオブジェクトに属する画像データが格納されているパケットであるとされれば、当該トランスポートパケットは、バッファTBn111Aに一旦溜め込まれ、所定のタイミングで読み出されて入力端103Bが選択されたスイッチ回路103に入力され、スイッチ回路103を介してコンテンツバッファ105に格納される。

#### 【0245】

PIDフィルタ110において、PIDに基づき、トランスポートパケットがサブピクチャデータすなわち字幕データが格納されているパケットであるとされれば、当該トランスポートパケットは、バッファTBn111BおよびバッファBn112Bに一旦溜め込まれ、所定のタイミングで読み出されて入力端107Bが選択されたスイッチ回路107に入力され、スイッチ回路107を介してサブピクチャデコーダ116に供給される。 20

#### 【0246】

サブピクチャデコーダ116では、供給されたトランスポートパケットのヘッダ情報を除去すると共に、当該トランスポートパケットに格納された字幕データがデコードされて字幕を表示するための画像データとされる。この画像データは、所定のタイミングでスイッチ回路130の入力端130Bに入力され、スイッチ回路130を介してサブピクチャプレーン132に展開される。

#### 【0247】

PIDフィルタ110において、PIDに基づき、トランスポートパケットが静止画像データが格納されているパケットであるとされれば、当該トランスポートパケットは、バッファTBn111CおよびバッファBn112Cに一旦溜め込まれ、所定のタイミングで読み出されて入力端108Bが選択されたスイッチ回路108に入力され、スイッチ回路108を介して静止画像デコーダ117に供給される。 30

#### 【0248】

静止画像デコーダ117では、供給されたトランスポートパケットのヘッダ情報を除去すると共に、当該トランスポートパケットに格納された静止画像データがデコードされ、静止画像データとされる。この静止画像データは、所定のタイミングでスイッチ回路131の入力端131Bに入力され、スイッチ回路131を介して静止画像プレーン134に展開される。 40

#### 【0249】

PIDフィルタ110において、PIDに基づき、トランスポートパケットがオーディオデータが格納されているパケットであるとされれば、当該トランスポートパケットは、バッファTBn111DおよびバッファBn112Dに一旦溜め込まれ、所定のタイミングで読み出されてオーディオデコーダ118に供給される。このトランスポートパケットに格納されるオーディオデータは、例えばMPEGに準拠した方式で圧縮符号化されている。

#### 【0250】

オーディオデコーダ118は、例えばリニアPCM(Pulse Code Modulation)オーディオ 50

オーディオデコーダ119を有し、入力されたトランスポートストリームのヘッダ情報を除去すると共に、当該トランスポートパケットに格納された圧縮符号化されたオーディオデータをリニアPCMオーディオデータにデコードする。

#### 【0251】

オーディオデコーダ118から出力されたりニアPCMオーディオデータは、オーディオ用のプレゼンテーションプロセッサ141に入力され、マルチメディアエンジン106の制御に基づき所定の音響効果などが付加されて、出力端142に導出される。

#### 【0252】

PIDフィルタ110において、PIDに基づき、トランスポートパケットが動画像データが格納されているパケットであるとされれば、当該トランスポートパケットは、バッファTBn111E、バッファMBn113およびバッファEBn114に一旦溜め込まれ、所定のタイミングで読み出されてMPEGビデオデコーダ120に供給される。このトランスポートパケットに格納される動画像データは、MPEG2方式により圧縮符号化されている。

#### 【0253】

MPEGビデオデコーダ120では、供給されたトランスポートパケットのヘッダ情報を除去すると共に、当該トランスポートパケットに格納された、MPEG2方式で圧縮符号化された動画像データをベースバンドの動画像データにデコードする。

#### 【0254】

MPEGデコーダ120から出力された動画像データは、スイッチ回路124の入力端124Aに入力されると共に、バッファ123を介してスイッチ回路124の入力端124Bに入力される。スイッチ回路124において、所定のタイミングで入力端124Aおよび124Bが選択され、出力された動画像データが動画像プレーン135に展開される。

#### 【0255】

PIDフィルタ110において、PIDに基づき、トランスポートパケットがシステム情報が格納されているパケットであるとされれば、当該トランスポートパケットは、バッファTBn111FおよびBsys115を介してシステムデコーダ121に供給される。システムデコーダ121では、供給されたトランスポートパケットのヘッダ情報が除去され、格納されているシステム情報が取り出される。システム情報は、例えば図示されないCPUに渡される。

#### 【0256】

サブピクチャプレーン132上の画像データは、上述のパレット22に対応するパレット150に供給され、256色からなるパレットに対してインデックスによる参照がなされ、YC bCrデータが出力されると共に、不透明度データ $\alpha 1$ が抜き出される。これらYC bCrデータおよび不透明度データ $\alpha 1$ は、プレゼンテーションプロセッサ139に供給される。

#### 【0257】

グラフィクスプレーン133上の画像データは、上述のRGB/YCbCr変換回路26に対応するRGB/YCbCr変換回路151に供給され、カラーシステムがRGB(4:4:4)からYC bCr(4:4:4)に変換される。RGB/YCbCr変換回路151から出力されたYC bCrデータは、プレゼンテーションプロセッサ139に供給される。それと共に、グラフィクスプレーン133上の画像データから不透明度データ $\alpha 2$ が抜き出され、プレゼンテーションプロセッサ139に供給される。

#### 【0258】

静止画像プレーン134上の画像データがスイッチ回路137の入力端137Aに入力される。また、動画像プレーン135上の動画像データがスイッチ回路137の入力端137Bに入力される。静止画像プレーン134上の画像データおよび動画像プレーン135上の動画像データは、M/Sスイッチプレーン136の制御により、スイッチ回路137で選択される。すなわち、静止画像プレーン134は、動画像プレーン135と同等に

扱われる。スイッチ回路137の出力は、アップ/ダウンコンバータ138を介してプレゼンテーションプロセッサ139に供給される。

【0259】

なお、アップ/ダウンコンバータ138は、画像の解像度を変換する回路であって、例えば高解像度のHD (High Definition)画像から通常の解像度を有するSD (Standard Definition)画像への変換を行う。

【0260】

プレゼンテーションプロセッサ139は、図17を用いて説明した、字幕プレーン11 (サブピクチャプレーン132) の画像データによる不透明度 $\alpha 1$ と、グラフィクスプレーン12 (グラフィクスプレーン133) による不透明度 $\alpha 2$ とを用いたアルファブレンディング処理を行う。 10

【0261】

すなわち、プレゼンテーションプロセッサ139では、静止画像プレーン134または動画像プレーン135のうちスイッチ回路137で選択された側の画像データに対して、サブピクチャプレーン132の画像データに設定された不透明度 $\alpha 1$ に基づき、サブピクチャプレーン132の画像データが合成される。さらに、静止画像プレーン134または動画像プレーン135およびサブピクチャプレーン132が合成された画像データに対して、グラフィクスプレーン133の画像データに設定された不透明度 $\alpha 2$ に基づき、グラフィクスプレーン133の画像データが合成される。この、グラフィクスプレーン133の画像データ、サブピクチャプレーン132の画像データ (字幕データ)、ならびに、静止画像プレーン134または動画像プレーン135の画像データが合成された画像データ 20

【0262】

なお、プレゼンテーションプロセッサ139は、画像データに対してリアルタイムでエフェクト処理を行うこともできる。

【0263】

上述では、プレーヤデコーダ100の各部がハードウェアで構成されるように説明したが、これはこの例に限られない。例えば、プレーヤデコーダ100をソフトウェア上の処理として実現することも可能である。この場合、プレーヤデコーダ100をコンピュータ装置上で動作させることができる。また、プレーヤデコーダ100をハードウェアおよび 30

ソフトウェアが混合された構成で実現することもできる。例えば、オーディオデコーダ118やMPEGビデオデコーダ120をハードウェアで構成し、その他をソフトウェアで構成することが考えられる。

【0264】

プレーヤデコーダ100をソフトウェアのみ、または、ハードウェアおよびソフトウェアの混合により構成し、コンピュータ装置で実行させるためのプログラムは、例えばCD-ROM (Compact Disc-Read Only Memory) といった記録媒体に記録されて提供される。このCD-ROMをコンピュータ装置のCD-ROMドライブに装填し、CD-ROMに記録されたプログラムを所定にコンピュータ装置にインストールすることで、上述の処理をコンピュータ装置上で実行可能な状態とすることができる。なお、コンピュータ装置の 40

【0265】

## 2-8. ユーザインターフェイス

次に、この発明の実施の第1の形態によるユーザインターフェイスについて、より詳細に説明する。図35は、グラフィクスプレーン12上に表示されるボタン表示の一例の状態遷移図である。ボタン表示の状態は、画面にボタンが表示されるボタン表示状態と、画面にボタンが表示されていないボタン非表示状態とに大別される。ボタン非表示状態からボタン表示が開始されボタン表示状態に遷移する。ボタン表示状態からボタン表示が終了され、ボタン非表示状態に遷移する。ボタン表示状態は、さらに、通常状態、選択状態および実行状態の3種類の状態を有する。ボタン表示は、これら3種類の状態間をそれぞれ 50

互いに遷移することができる。遷移方向を一方向に限定することも可能である。

#### 【0266】

上述の図20を用いて、より具体的に説明する。先ず、ディスク挿入時や、ユーザがリモコンのメニューキーを押したときに、メニュー画面60が表示される。メニュー画面60の表示に伴い、ボタン62A、62B、62C、63D、64および65が非表示状態から表示状態へと遷移する。ボタン表示開始時は、ボタン62A、62B、62C、63D、64および65のうち一つが予め選択状態にされていることが多い。例えばボタン62Aが予め選択状態とされ、他のボタンが通常状態とされているものとする。

#### 【0267】

この状態で、ユーザがリモコンの例えば矢印キーを操作すると、通常状態のボタンの一つ（例えばボタン62B）が選択状態に遷移すると共に、選択状態だったボタン62Aが通常状態へと遷移する。このようにして、ユーザ操作によってカーソルが移動していく。ユーザがリモコンの決定キーを操作すると、そのときに選択状態にあったボタン62Bが選択状態から実行状態へと遷移し、当該ボタン62Bに予め割り当てられているプレーヤ動作が実行される。

#### 【0268】

上述したように、プレーヤ動作は、プログラミング言語やECMAScriptなどのスクリプト言語により記述され、ディスク上に記録され用意されている。このプレーヤ動作に関する記述は、独立したファイルとしてディスク上に記録してもよいし、後述するグラフィックオブジェクトのように、クリップAVストリームファイルに多重化することも可能である。

#### 【0269】

### 2-9. データ構造について

次に、このようなメニュー画面を構成するボタンなどの画像データおよび当該画像データに付随する制御情報のデータ構造について説明する。ここでは、ディスクに収録されるコンテンツ本体をなす動画以外に表示される映像要素として、字幕およびグラフィクス（静止画）を考える。以下では、画面に表示される字幕やグラフィクスといった要素をオブジェクトとして考え、オブジェクトの種類を、図36に示されるように、字幕、同期型グラフィクスおよび非同期型グラフィクスの3種類に分類する。

#### 【0270】

字幕とは、例えば映画の字幕のように、動画と同期して表示／非表示の切り換えが行われ、リモコンなどによるユーザ入力とは無関係な映像要素を指す。グラフィクスとは、メニュー画面上のボタンなどのように、ユーザ入力を受け付けることができる映像要素とする。さらに、グラフィクスは、同期型グラフィクスおよび非同期型グラフィクスの2種類に分類される。同期型グラフィクスとは、例えば本編再生中のあるタイミングに表示される分岐選択画面のように、動画と同期して表示される映像要素とする。また、非同期型グラフィクスとは、ディスク挿入時に最初に表示されるメニュー画面や、ユーザ入力に応じて表示される画面など、本編の再生とは非同期に表示される映像要素を指す。JavaVM上で動作するJavaアプリケーションにより表示される映像要素や、ブラウザソフトウェア上でHTMLファイルの記述に基づき表示される映像要素は、非同期グラフィクスである。

#### 【0271】

各映像要素と動画プレーン10に表示される本編映像との関係からいえば、字幕および同期型グラフィクスは、本編映像に同期して表示されるため、共に同期型である。一方、非同期型グラフィクスは、上述のように本編映像と非同期で表示されるので、その名の示す通り、非同期型である。

#### 【0272】

また、字幕とグラフィクスとは、表示先のプレーンの違いから分類することができる。字幕は、字幕プレーン11に表示される。同期型および非同期型グラフィクスは、グラフィクスプレーン12に表示される。

## 【0273】

ここで、字幕および同期型グラフィクスは、本編の動画再生中に表示するという共通の性質を持つことから、共通のデータ構造とすることが好ましい。以下では、これら共通のデータ構造とされる字幕および同期型グラフィクスを、グラフィクスオブジェクトと称する。グラフィクスオブジェクトは、常に動画再生と同期して表示されるため、動画と同一のストリームに多重化しておく、扱いが容易になる。

## 【0274】

図37は、この発明の実施の第1の形態によるグラフィクスオブジェクト200のデータ構造の例を示す。グラフィクスオブジェクト200は、図37Aに一例が示されるように、グラフィクスオブジェクトヘッダ201、表示制御命令テーブル202およびPNGデータ領域203の3つの領域からなる。

## 【0275】

なお、以下の例では、グラフィクスオブジェクト200で扱われる画像データの形式をPNG形式とし、画像データをPNG画像データとしているが、JPEG形式やビットマップデータ、ランレングス圧縮された画像データ、圧縮符号化がなされていないビットマップデータなど、他の形式の画像データをグラフィクスオブジェクト200で扱うことも可能である。ここでは、便宜上、画像データをPNG画像、PNG画像データなどと表現する。

## 【0276】

図37において、グラフィクスオブジェクトヘッダ201は、当該グラフィクスオブジェクト200の属性を表す情報が格納される。グラフィクスオブジェクト200の属性は、例えば、当該グラフィクスオブジェクト200のデータサイズ、当該グラフィクスオブジェクト200が有するPNG画像数、当該グラフィクスオブジェクト200が有するPNG画像データが共通に使用するパレットデータ、ならびに、当該グラフィクスオブジェクト200を識別する識別情報からなる。識別情報は、例えばグラフィクスオブジェクト200毎に一意に割り当てられた番号である。グラフィクスオブジェクトヘッダ201に、さらに他の情報を格納してもよい。

## 【0277】

表示制御命令テーブル202は、当該グラフィクスオブジェクト200が有するPNG画像の表示位置、表示開始時刻および表示終了時刻といった、PNG画像の表示を制御するための情報が格納されるテーブルである。

## 【0278】

PNGデータ領域203は、PNG形式で圧縮符号化された画像データ（以下、PNGデータと称する）が格納される。PNGデータ領域203には、複数のPNGデータ203A、203B、・・・、203nを格納することができる。PNGデータ領域203に格納されるPNGデータ203の数は、上述したグラフィクスオブジェクトヘッダ201に記述されている。

## 【0279】

なお、PNGデータ領域203に格納される複数のPNGデータ203A、203B、・・・、203nは、アニメーションを構成する複数枚の静止画の組や、上述したボタン表示の3状態を表す画像といった、相互に関連の強い画像であることを想定している。これらを、一つのグラフィクスオブジェクト200に纏めることで、PNG画像の扱いを容易にすることができる。

## 【0280】

グラフィクスオブジェクト200は、表示が可能になる時刻を示す時刻情報を有する。リアルタイムストリームがMPEG2 TSで伝送されるこの例では、この時刻情報として、MPEG2 (Moving Pictures Experts Group 2)により定義された、pts (Presentation Time Stamp)を用いる。ptsは、再生出力の時刻管理情報で、90kHzのクロックで計測した値を33ビット長で表す。MPEGシステムの基準復号器内部のSTC (System Time Clock)がptsに一致したときに、対象となるアクセスユニットを再生出力す

る。一つのグラフィクスオブジェクト200は、ptsで表される時刻から表示が可能になり、その時刻以降に、表示制御命令により実際に表示のON/OFFが行われる。表示制御命令により表示の管理を行うため、表示を停止した後に再度、同じグラフィクスオブジェクト200を表示させるような制御が可能である。

#### 【0281】

図37Bは、字幕を格納するグラフィクスオブジェクト200の例を示す。PNGデータ領域203に、字幕として表示する画像がPNGデータ(1)203A-1として格納される。通常、字幕であれば、一つのグラフィクスオブジェクト200に含まれるデータは、PNGデータ(1)203A-1一つで十分である。字幕に対して、フェードイン/フェードアウトのような、画像の内容が変化しない特殊効果を施すときは、表示制御命令テーブル202に対してPNGデータ(1)A-1の不透明度を変化させる表示制御命令を加えることで、実現できる。

10

#### 【0282】

一方、字幕に対して、アニメーションなどのような、画像そのものが変化するような効果を施す場合には、図37Bに点線で示されるように、アニメーションの各動作に対応する複数のPNGデータ(2)B-1、PNGデータ(3)C-1、PNGデータ(4)D-1、・・・を、一つのグラフィクスオブジェクト200に格納すればよい。これに限らず、日本語字幕、英語字幕など、言語が異なる字幕を表示するPNGデータをPNGデータ(1)A-1、PNGデータ(2)B-1、・・・として、一つのグラフィクスオブジェクト200に格納することもできる。

20

#### 【0283】

図37Cは、ボタンを構成するグラフィクスオブジェクト200の例を示す。ボタンは、上述したように、通常状態、選択状態および実行状態の3種類の状態を有し、これら3種類の状態のそれぞれでボタン表示を異ならせることができる。3種類の状態でボタン表示を異ならせる場合、3枚のボタン画像データを用意しておく必要がある。これら3枚のボタン画像データを、ひとまとまりのグラフィクスオブジェクト200として扱う。すなわち、グラフィクスオブジェクト200のPNGデータ領域203に、通常状態、選択状態および実行状態それぞれのボタン表示に用いるPNGデータ203A-2、203B-2および203C-2を格納する。

#### 【0284】

図37Bに実線で示される通常の字幕のように、グラフィクスオブジェクト200内に一つだけ字幕表示のPNGデータ203A-1が存在する場合には、当該グラフィクスオブジェクト200内の表示制御命令テーブル202に格納される表示制御命令は、そのPNGデータ203A-1に対するものとなる。一方、図37Cに一例が示されるような、グラフィクスオブジェクト200内に、複数のPNGデータ203A-2、203B-2および203C-2が存在する場合、表示制御命令テーブル202に格納される表示制御命令が、複数のPNGデータ203A-2、203B-2および203C-2のうちのどのデータに対する表示制御命令であるかを特定する必要がある。

30

#### 【0285】

例えば、図37Cに例示される、ボタンのグラフィクスオブジェクト200においては、このボタンの表示開始時の初期状態が選択状態と定められている場合は、最初に表示するボタン画像は、PNGデータ領域203の先頭に配置される、通常状態時のPNGデータ203A-2ではなく、選択状態時のPNGデータ203B-2としなければならない。この発明の実施の第1の形態では、このような表示制御は、グラフィクスオブジェクト200の外で行うようにしている。

40

#### 【0286】

例えば、各ボタンの初期状態や、表示開始および表示停止、ボタンが実行状態に移された際に実行されるプログラムなどは、グラフィクスオブジェクト200外部のスクリプトプログラム、例えば上述のECMAScriptやJavaScriptによって指定する方法が考えられる。また、ボタンの表示中における、ボタン表示を行うPNGデータの

50

変更は、例えばユーザがリモコンの矢印キーを操作し、カーソルを移動させた際に発生する。この場合には、プレーヤがユーザ入力に応じて各ボタンのPNGデータを入れ替えることになる。

#### 【0287】

この発明の実施の第1の形態では、グラフィクスオブジェクト200は、MPEG2に規定されるパケットに分割され、クリップAVストリームに多重化され、クリップAVストリームファイルとしてディスクに記録される。図38に一例が示されるように、グラフィクスオブジェクト200は、分割され、MPEG2に規定されるPES(Packetized Elementary Stream)パケット210、210、・・・に格納される。このとき、グラフィクスオブジェクトヘッダ201および表示制御命令テーブル202、ならびに、各PNGデータ203A、203B、・・・、203nそれぞれの先頭が、PESパケット210、210、・・・のペイロードの先頭に合致するように、データが配置される。こうすることで、再生時に、グラフィクスオブジェクト200内の各データの検索が容易となる。

#### 【0288】

こうして分割されてPESパケット210、210、・・・に格納されたグラフィクスオブジェクト200は、さらに、データサイズが188バイトに固定されたTSパケットに分割され(図示しない)、クリップAVストリームなどの動画データや音声データによるストリームに多重化される。

#### 【0289】

2-10. グラフィクスオブジェクトのデコーダモデルについて

図39は、グラフィクスオブジェクト200のデコードを行うグラフィクスオブジェクトデコーダモデル240の一例の構成を示す機能ブロック図である。このグラフィクスオブジェクトデコーダモデル240は、図34を用いて説明したプレーヤデコーダ100におけるマルチメディアエンジン106およびサブピクチャデコーダ116を中心に構成される。図39において、図34と共通する部分には同一の符号を付して詳細な説明を省略する。また、この図39は、より機能面に注目して表現されており、上述の図34とは異なる表現がなされている場合があるので、必要に応じて図34の構成と対応付けながら説明する。

#### 【0290】

端子202からMPEG TSとして入力されたクリップAVストリームがPIDフィルタ110に供給される。PIDフィルタ110は、MPEG2 TS(トランスポートストリーム)に対するデマルチプレクサとして機能し、供給されたMPEG TSのPIDに基づき、動画データ、オーディオデータおよびグラフィクスオブジェクト200をそれぞれ抜き出す。動画データは、ビデオバッファとしてのバッファTBn111Eに供給される。オーディオデータは、オーディオバッファとしてのバッファ111Dに供給される。グラフィクスオブジェクト200は、グラフィクスオブジェクト(図39では「GOBJ」と表記)の入力バッファであるバッファTBn111Bに供給される。

#### 【0291】

グラフィクスオブジェクト200は、バッファTBn111Bから読み出され、GOBJパーサ224に供給される。GOBJパーサ224は、例えば図34におけるサブピクチャデコーダ116の機能の一つである。GOBJパーサ224では、供給されたグラフィクスオブジェクト200のグラフィクスオブジェクトヘッダ201を読み取り、パレットデータを抽出すると共に、表示制御命令テーブル202およびPNGデータ領域203を分離する。パレットデータおよび表示制御命令テーブル202は、コマンドプロセッサ/グラフィックレンダラ225に供給される。また、PNGデータ領域203のPNGデータ203A、203B、・・・は、PNGデコーダバッファ226に一時的に格納される。PNGデコーダバッファ226は、例えば図34におけるバッファBn112Bに対応する。

#### 【0292】

PNGデコーダバッファ226に格納されたPNGデータ203は、サブピクチャデコ



ーダ116の機能の一つであるPNGデコーダ227によりデコードされ、ビットマップデータとされる。このビットマップデータは、オブジェクトバッファ228に蓄積される。オブジェクトバッファ228は、図34におけるグラフィクスバッファ109Dに対応する。

#### 【0293】

コマンドプロセッサ／グラフィックレンダラ225は、GOBJパーサ224から供給された表示制御命令テーブル202に記述される表示制御命令に従い、オブジェクトバッファ228に蓄積されているビットマップデータを読み出して指定された時刻にプレーンバッファ229に転送する。プレーンバッファ229は、例えば図34におけるサブピクチャプレーン132およびグラフィクスプレーン133に対応させることができる。例えば、字幕と、字幕以外のグラフィクスオブジェクトとに対してそれぞれプレーンバッファ229A、229B（図示しない）を設けることができる。これに限らず、サブピクチャプレーン132およびグラフィクスプレーン133を、プレーンバッファ229上の異なる領域とすることもできる。

#### 【0294】

また、コマンドプロセッサ／グラフィックレンダラ225は、GOBJパーサ224から供給されたパレットデータを、図34のパレット150に対応する共通パレットテーブル230に供給する。コマンドプロセッサ／グラフィックレンダラ225は、図34のマルチメディアエンジン106の一部の機能およびサブピクチャデコーダ116の一部の機能を併せ持つ。

#### 【0295】

グラフィクスオブジェクト200がボタンを構成するものである場合、上述したように、ボタンの3種類の状態にそれぞれ対応したPNGデータ203A、203Bおよび203Cがグラフィクスオブジェクト200に格納される。PNGデータ203A、203Bおよび203Cは、PNGデコーダ227でデコードされ、オブジェクトバッファ228にそれぞれ蓄積される。

#### 【0296】

一方、ユーザのリモコンなどによる入力が、コマンドプロセッサ／グラフィックレンダラ225により受け取られる。コマンドプロセッサ／グラフィックレンダラ225は、このユーザ入力に応じて、オブジェクトバッファ228から該当するビットマップを読み出して、プレーンバッファ229に転送する。例えば、ユーザ入力があるボタンに対して選択状態から実行状態に状態遷移させるものであれば、オブジェクトバッファ228から、実行状態のボタン画像に対応するビットマップデータが選択的に読み出されて、プレーンバッファ229に転送される。

#### 【0297】

なお、コマンドプロセッサ／グラフィックレンダラ225は、オブジェクトバッファ228から読み出したビットマップデータに対して、表示制御命令に従い部分切り出しなどの特殊効果処理を施すこともできる。

#### 【0298】

この実施の第1の形態では、PNGデータは、1画素のサンプリング深さが8ビットとされているため、プレーンバッファ229は、1画素当たり8ビットのデータが並んでいることになる。プレーンバッファ229は、ディスプレイ装置などに実際に画像を表示させる処理を行う表示系の走査周期毎に読み出される。プレーンバッファ229から読み出されたビットマップデータは、例えば図34のパレット150に対応できる共通パレットテーブル230に供給され、パレットインデックス値から実際のRGB（4：4：4）の色情報に変換されると共に、不透明データ $\alpha 1$ 、 $\alpha 2$ が抜き出される。RGB（4：4：4）の色情報は、図示されない変換回路でYCbCr（4：4：4）の色情報に変換され、不透明データ $\alpha 1$ 、 $\alpha 2$ と共に、図34のプレゼンテーションプロセッサ139に供給される。

#### 【0299】

なお、例えばフェードイン／フェードアウトなどの、パレットや不透明度を変える処理が必要な特殊効果は、コマンドプロセッサ／グラフィックレンダラ 225 が表示制御命令に従い共通パレットテーブル 230 のデータを変化させることで実現される。また、字幕と、字幕以外のグラフィクスオブジェクト 200 に対して、それぞれ共通パレットテーブル 230A、230B (図示しない) を設けてもよい。

#### 【0300】

図40は、グラフィクスオブジェクト入力バッファ (バッファTBn111B、以下GOBJバッファ)、PNGデコーダ227、オブジェクトバッファ228およびプレーンバッファ229における蓄積データ量の遷移の例を概略的に示す。なお、PNGデコーダ227は、PNGデコーダ227がPNGデータのデコード時に用いるバッファにおける蓄積データ量が示されている。 10

#### 【0301】

図40の例では、3つのグラフィクスオブジェクトGOBJ#1、GOBJ#2およびGOBJ#3によるデータ量の遷移が時間の経過に従い示される。グラフィクスオブジェクトのデコード開始時刻は、MPEG2システムにおけるdts (Decoding Time Stamp) で表される。また、オブジェクトの有効期間は、開始時刻がptsで表され、グラフィクスオブジェクトヘッダ201内に記述される時刻presentation\_endで終了される。この有効期間内に、表示制御命令によって画像の表示開始および終了が指示される。

#### 【0302】

図40Dを参照し、GOBJバッファにグラフィクスオブジェクトGOBJ#1のPNGデータが入力され、時刻dts of GOBJ#1に、このPNGデータのデコードが開始される。図40Cを参照し、PNGデータがGOBJ入力バッファからPNGデコーダ227に転送され、PNGデータがビットマップデータにデコードされる。実際には、PNGデータは、GOBJ入力バッファから一旦PNGデコーダバッファ226に溜め込まれ、PNGデコーダ227は、PNGデコーダバッファ226に溜め込まれたデータを用いてデコード処理を行う。 20

#### 【0303】

PNGデコーダ227は、デコード速度に上限があるため、GOBJ入力バッファからPNGデコーダバッファ226に対して、PNGデコーダ227のデコード速度を超えないような転送速度でデータが供給される。そのため、PNGデコーダバッファ226に対して、PNGデコーダ227での処理時間を0としたときの垂直線に対し、ある傾きAに対応するデータ転送レートで以てPNGデータが入力される。 30

#### 【0304】

PNGデータのPNGデコーダ227への入力が完全に終了していなくても、PNGデータのデコードを開始することが可能である。また、この図40の例では、GOBJ入力バッファに格納されたオブジェクトGOBJ#1が全てPNGデコーダ227に転送された後、次のオブジェクトGOBJ#2のPNGデータの、GOBJバッファへの入力が開始されている。

#### 【0305】

なお、オブジェクトGOBJ#2およびオブジェクトGOBJ#3についても、ある傾きBおよびCにそれぞれ対応する転送レートで以て、PNGデコーダバッファ226に対してPNGデータが入力される。傾きBは、実際には、複数の区間でそれぞれ異なる傾きとなっている。 40

#### 【0306】

時刻pts of GOBJ#1によりオブジェクトGOBJ#1の有効期間が開始されると、デコードされPNGデコーダバッファに蓄積された、オブジェクトGOBJ#1のビットマップデータがオブジェクトバッファ228に転送される (図40B)。オブジェクトバッファ228に転送されたオブジェクトGOBJ#1は、このオブジェクトGOBJ#1に対して時刻presentation\_end of GOBJ#1で示される時刻まで、有効期間が持続される。

#### 【0307】

オブジェクトGOBJ#1の有効期間内に、命令Display ON cmd.of GOBJ#1によりオブジェクトGOBJ#1の表示命令が出されたら、オブジェクトバッファ228に蓄積されたオブジェクトGOBJ#1のビットマップデータがプレーンバッファ229に転送され、表示される(図40A)。詳細は後述するが、ビットマップデータをプレーンバッファ229へ転送する際の転送レートは、バス幅などの影響により上限を有する。そのため、プレーンバッファ229に対するビットマップデータの書き込みは、例えばある傾きDに対応する転送レートで以て行われる。

#### 【0308】

これは、他のオブジェクトGOBJ#2やオブジェクトGOBJ#3についても同様に、ある傾きE、F、Gに対応する転送レートで以てビットマップデータが転送され、プレーンバッファ229に書き込まれる。

#### 【0309】

表示は、オブジェクトGOBJ#1に対して表示停止を命令するコマンドDisplay OFF cmd.of GOBJ#1が出されるまで、持続される。コマンドDisplay OFF cmd.of GOBJ#1が出されると、プレーンバッファ229に格納された当該オブジェクトGOBJ#1のビットマップデータが破棄され、表示が停止される。

#### 【0310】

GOBJバッファには、オブジェクトGOBJ#2およびGOBJ#3のPNGデータも、順次、入力される。そして、オブジェクトGOBJ#1の場合と同様に、時刻dts of GOBJ#2、時刻dts of GOBJ#3からそれぞれデコード開始が指示され、PNGデータがPNGデコーダ227に供給され、PNGデコーダバッファを用いてビットマップデータへのデコードが行われる。そして、オブジェクトGOBJ#2に対して時刻pts of GOBJ#2により有効期間開始が指示され、図40では省略されているが、コマンドDisplay ON cmd.of GOBJ#2によりオブジェクトGOBJ#2の表示命令が出され、オブジェクトバッファ228からプレーンバッファ229にビットマップデータが転送され、コマンドDisplay OFF cmd.of GOBJ#2が出されるまで、オブジェクトGOBJ#2が表示される。

#### 【0311】

この図40の例では、オブジェクトGOBJ#2は、一旦、図示されないコマンドDisplay OFF cmd.of GOBJ#2により表示が停止された後、再びコマンドDisplay ON cmd.of GOBJ#2による表示が再開されている。オブジェクトバッファ228に格納されたオブジェクトGOBJ#2のビットマップデータは、オブジェクトGOBJ#2に対して有効期間終了時刻presentation end of GOBJ#1を指定するまで保持されるため、コマンドDisplay ON cmd.of GOBJ#2を与えることにより、繰り返しオブジェクトGOBJ#2を表示させることができる。

#### 【0312】

一方、オブジェクトGOBJ#3は、オブジェクトGOBJ#2の有効期間中に重複して、有効期間が指定されている。この場合、オブジェクトバッファ228は、空き容量に応じて、複数のビットマップデータを互いに異なる領域に格納する。例えば、オブジェクトバッファ228からオブジェクトGOBJ#2のビットマップデータをプレーンバッファ229に転送して表示中に、オブジェクトバッファ228の異なる領域からオブジェクトGOBJ#3のビットマップデータをプレーンバッファ229に転送することで、2枚のビットマップデータを同時に表示させることができる。

#### 【0313】

##### 2-1-1. グラフィックスの転送速度について

ここで、このグラフィックスオブジェクトデコーダモデル240(以下、デコーダモデル240と略称する)をプレーヤに実装する場合について考える。同一ディスクを異なるプレーヤで再生した際の、プレーヤ同士の再生互換性を保つためには、デコーダモデル240に対して所定の制限を設けることが必要となる場合がある。例えば、デコーダモデル240のグラフィックス処理能力には上限があり、その能力を超えるグラフィックスデータが入力された場合には、グラフィックスデータのデコードを完全に行うことができなくなり、正

常な表示を行うことができなくなる。

【0314】

すなわち、プレーヤ側では、プレーヤが最小限備える必要のあるグラフィクス処理能力を規格により定める。一方、ディスク側では、ディスクに記録するコンテンツを制作する側において、規格により定められたプレーヤの能力の範囲内で処理可能なグラフィクスを用意する。このようにしてプレーヤ側とディスク制作側とでグラフィクス処理能力の整合性を図ることで、再生互換性を保つことができる。

【0315】

この発明の実施の第1の形態では、上述の図39における、GOBJパーサ224からPNGデコーダバッファ226へのデータ転送速度R(1)と、コマンドプロセッサ225からプレーンバッファ229へのデータ転送速度R(2)とを規定する。

【0316】

データ転送速度R(1)は、PNGデコーダバッファ226に入力されるデータの、単位時間当たりのデータ転送量を規定する。すなわち、上述の図40Cに示される傾きA、BおよびCがデータ転送速度R(1)に対応する。これは、PNGデコーダバッファ226の後に接続されるPNGデコーダ227の、圧縮符号化されたグラフィクスデータを単位時間当たりにデコードできる量を示すデコード能力を規定する。したがって、データ転送速度R(1)を制限することで、入力された圧縮符号化グラフィクスデータに対してデコードが間に合わなくなり、表示などが破綻状態になる状況が防止される。

【0317】

データ転送速度R(2)は、画像の更新速度を規定する。プレーンバッファ229は、実際に表示装置に表示される画面に対応している。そのため、プレーンバッファ229へのデータの書き込み速度によって、ユーザが見るグラフィクスの更新速度が決まる。データ転送速度R(2)の単位は、[バイト/秒]であって、プレーン全体すなわち全画面を更新する際の最小更新時間間隔を規定する。すなわち、上述の図40Aに示される傾きD、E、FおよびGがデータ転送速度R(2)に対応する。

【0318】

プレーンの一部だけを更新する場合には、更新される画像データが少ないためデータ転送速度R(2)で規定される最小更新時間間隔よりも短い周期で更新される。但し、更新間隔は、更新される画像データのデータ量に反比例するとは限らず、当該画像データのプレーン上の配置により大きく影響される。

【0319】

プレーンの更新速度について、図41を用いてより具体的に説明する。オブジェクトバッファ228には、2個のグラフィクスオブジェクト460および461が格納されているものとする。ここで、これら2個のグラフィクスオブジェクト460および461を、同時にプレーンバッファ229に書き込み表示させることを考える。

【0320】

グラフィクスオブジェクト460および461は、オブジェクトバッファ228から読み出され、コマンドプロセッサ/グラフィックレンダラ225に供給される。このコマンドプロセッサ/グラフィックレンダラ225からの出力が上述のデータ転送速度R(2)により制限され、画面上の更新速度(更新間隔)が制限されることになる。

【0321】

このとき、更新速度は、グラフィクスオブジェクト460および461の合計のデータ量ではなく、グラフィクスオブジェクト460および461を、プレーン上にどのように配置するかによって決まる。これは、グラフィクス処理を行う一般的なグラフィクスプロセッサでは、プレーンの更新を矩形領域で行うためである。

【0322】

例えば、プレーン上に配置されるグラフィクスオブジェクト460および461を全て含む矩形領域262で以て、プレーンの更新がなされる。すなわち、コマンドプロセッサ/グラフィックレンダラ225において、グラフィクスオブジェクト460および461

の配置情報に基づき矩形領域 2 6 2 の画像データが形成される。この矩形領域 2 6 2 の画像データが例えばバスを介してプレーンバッファ 2 2 9 に供給される。プレーンバッファ 2 2 9 では、例えば表示位置の指定に基づき、矩形領域 2 6 2 に対応する領域のデータが矩形領域 2 6 2 のデータと置き換えられる。

#### 【0 3 2 3】

コマンドプロセッサ／グラフィックレンダラ 2 2 5 から出力される画像データは、ビットマップデータであるので、画像の内容によらず、画像の面積に応じたデータ量を有することになる。図 4 1 の例の場合、2 個のグラフィクスオブジェクト 4 6 0 および 4 6 1 を含む矩形領域 2 6 2 の画像データ量は、例えば [幅(Width)×高さ(Height)] バイトで表すことができる。

10

#### 【0 3 2 4】

ここで、プレーンバッファ 2 2 9 へのデータ転送速度が速度  $R(2)$  [バイト/秒] とされていることから、これら 2 個のグラフィクスオブジェクト 4 6 0 および 4 6 1 は、 $\{ \text{速度 } R(2) / (\text{幅} \times \text{高さ}) \}$  秒の時間で更新できることが分かる。すなわち、 $\{ \text{速度 } R(2) / (\text{幅} \times \text{高さ}) \}$  秒後には、次のグラフィクスオブジェクトの描画が可能となる。したがって、ディスク制作側では、次のグラフィクスオブジェクトの描画を、少なくともこの時間以上の間隔を空けて行うようにプログラムすることで、どのプレーヤでも同一のグラフィクス表示がなされ、再生互換性を保つことができる。

#### 【0 3 2 5】

後述する動きのある字幕表示についても、上述のようにしてデータ転送速度  $R(2)$  を見積もることで、再生互換性を保つようにアニメーション速度を決めることができる。

20

#### 【0 3 2 6】

##### 2-12. グラフィクスオブジェクトの構造の詳細

次に、グラフィクスオブジェクト 2 0 0 の構造について、より詳細に説明する。図 4 2 は、グラフィクスオブジェクト 2 0 0 の一例の構造を表すシンタクスを示す。図 3 7 A などにおけるグラフィクスオブジェクトヘッダ 2 0 1 がブロック GraphicsObjectHeader()、表示命令制御テーブル 2 0 2 がブロック GOBJCommandTable()、PNG データ領域 2 0 3 がブロック PNGImageRegion() にそれぞれ相当する。

#### 【0 3 2 7】

ブロック GraphicsObjectHeader() の先頭にフィールド length が配置される。フィールド length は、データ長が 8 ビットの 0 以上の整数で、フィールド length の直後からブロック GraphicsObjectHeader() の終わりまでの長さをバイトで示す。フィールド presentation\_end\_time\_stamp は、データ長が 3 3 ビットの 0 以上の整数で、このグラフィクスオブジェクト 2 0 0 の有効期間終了時刻を表す。グラフィクスオブジェクトの有効期間は、PES パケットヘッダの pts からこのフィールド presentation\_end\_time\_stamp で示される有効期間終了時刻までである。フィールド Number\_of\_PNG\_images は、データ長が 8 ビットの 0 以上の整数で、ブロック PNGImageRegion() に格納されている PNG 画像の数を表す。フィールド Number\_of\_DispatchCmds は、データ長が 8 ビットの 0 以上の整数で、ブロック GOBJCommandTable() に格納されている表示制御命令の数を表す。

30

#### 【0 3 2 8】

ブロック GraphicsObjectHeader() 内のブロック GlobalPaletteTable() は、このグラフィクスオブジェクト 2 0 0 で共通に使用されるパレットテーブルの情報が格納される。例えばこのブロック GlobalPaletteTable() に記述されるパレットテーブルの情報が、例えば共通パレットテーブル 2 3 0 の内容としてセットされる。フィールド start\_address\_of\_PNG\_image(i) は、データ長が 3 2 ビットの 0 以上の整数で、i 番目の PNG 画像のデータ PNG\_image(i) が開始される位置を、この GraphicsObject() の先頭からの相対バイト数で表現する。

40

#### 【0 3 2 9】

フィールド PNG\_file\_name(i) は、フィールド start\_address\_of\_PNG\_image(i) から始まる PNG データのファイル名を表す。ブロック PNGImageRegion() 内のフィールドであるフ

50

フィールドPNG\_image(i)の内容は、単体のPNGファイルと同一であり、ブロックPNGImageRegion()は、一つ以上のPNGファイルを連結して作られる。例えば、上述の図37Aでは、PNGデータ203A、203B、・・・、203nが連結されてブロックPNGImageRegion()が作られる。その際にファイル名が失われないよう、フィールドPNG\_file\_name(i)にファイル名を保持しておくことができる。逆に、ブロックPNGImageRegion()を分解して個別のPNGファイルを復元する際には、各フィールドPNG\_image(i)を、iが対応するフィールドPNG\_file\_name(i)で示されるファイル名の、独立したファイルとすればよい。

#### 【0330】

ブロックGOBJCommandTable()は、実行時刻が同じ表示制御命令を集めた命令グループDispCmds(i)から構成される。命令グループDispCmds(i)は、必ず実行時刻を指定する命令execution\_time(time)から開始して、表示制御命令が並べられる。換言すると、命令execution\_time(time)から次の命令execution\_time(time)の前までが一つの命令グループDispCmds(i)を構成している。

#### 【0331】

ブロックPNGImageRegion()は、上述したように、PNG方式で圧縮符号化された画像一枚分のデータであるフィールドPNG\_image(i)の並びで構成される。

#### 【0332】

なお、ブロックGraphicsObjectHeader()とブロックGOBJCommandTable()との間には、任意の数のpadding\_wordを挿入することができる。同様に、ブロックGOBJCommandTable()とブロックPNGImageRegion()の間には任意の数のpadding\_wordを挿入することができる。

#### 【0333】

図43は、上述したブロックGlobalPaletteTable()の一例の構造を表すシンタクスを示す。フィールドnumber\_of\_palette\_entriesは、次に羅列されているパレットデータの数を表す。データ長が8ビットのインデックス番号で画像を表現する場合、このフィールドnumber\_of\_palette\_entriesの値は、最大で256となり、256色を用いることができる。この256色のうち一部の色だけを使用する場合、必要なパレットデータだけが存在すればよい。このフィールドnumber\_of\_palette\_entriesにより、用いるインデックスの数が示される。

#### 【0334】

フィールドpalette\_index\_numberは、次のフィールドred\_value、フィールドgreen\_valueフィールドblue\_valueフィールドalphaでそれぞれ定義される色や不透明度に割り当てられたインデックス番号を表す。画像データからは、このインデックス番号によって色や不透明度 $\alpha$ が参照される。

#### 【0335】

このブロックGlobalPaletteTable()においてfor文で示されるループの中では、同じ値を持つフィールドpalette\_index\_numberが二度以上現れてはならない。フィールドred\_value、フィールドgreen\_valueおよびフィールドblue\_valueは、それぞれデータ長が8ビットの0以上の整数であって、それぞれ赤、緑、青の色を指定する。8ビットのデータ長を有するフィールドalphaは、不透明度 $\alpha$ を表し、値0が完全に透明、値255が完全に不透明な状態を表す。

#### 【0336】

なお、各PNG画像は、パレット情報PLTEを格納するチャンクを持つことができるが、この実施の第1の形態では、そのパレット情報PLTEを使用せず、ブロックGlobalPaletteTable()で定義されたパレット情報を使うものとする。これは、同時に複数のPNG画像を表示する場合、各PNG画像が異なるパレットテーブルに基づく色を使っていると、それぞれ正しい色で表示することが困難になるためである。すなわち、GraphicsObject()に属する、フィールドPNG\_image(i)でそれぞれ表される複数のPNG画像は、全て共通のブロックGlobalPaletteTable()を参照、ブロックGlobalPaletteTable()に示される共通のパレットテーブルを用いるものとする。

#### 【0337】

次に、命令グループDispCmds(i)について説明する。命令グループDispCmds(i)は、グラフィクスオブジェクト200による表示を制御する表示制御命令が羅列される。命令グループDispCmds(i)において、命令execution\_time(start\_time)は、次の命令execution\_time(start\_time)の前までの命令を、指定の時刻start\_timeに実行することを指示する。時刻start\_timeの原点は、当該グラフィクスオブジェクト200のptsとする。また時刻start\_timeの単位は、ptsと同じものが用いられる。

#### 【0338】

一つの命令グループDispCmds(i)は、命令execution\_time(start\_time)により時刻start\_timeから始まり、その他の複数の命令を置くことができる。命令グループDispCmds(i)に置かれたこれらの命令は、命令execution\_time(start\_time)で指定される時刻start\_timeに同時に、実行されるものとする。また、命令グループDispCmds(i)による命令の実行完了時刻前に、次の命令グループDispCmds(i+1)の命令execution\_time(start\_time)で示される時刻start\_timeになった場合は、命令グループDispCmds(i)の実行は中断され、次の命令グループDispCmds(i+1)が実行される。

#### 【0339】

命令グループDispCmds(i)に置かれる、命令execution\_time(start\_time)以外の表示制御命令としては、図44に一覧が例示されるように、以下のものが考えられる。なお、各行頭の番号は、図44の各番号にそれぞれ対応している。(すなわち、番号(1)を、命令execution\_time(start\_time)とする)

- (2) グラフィクスオブジェクトの表示を開始する命令
- (3) グラフィクスオブジェクトの表示を停止する命令
- (4) 使用中のパレットテーブルの色や不透明度を変更する命令
- (5) グラフィクスオブジェクトのプレーン上での表示位置やサイズを設定する命令
- (6) グラフィクスオブジェクトの中の表示範囲を設定する命令
- (7) 効果音の再生を行う命令
- (8) 画像データ(PNGデータ)に効果音を割り当てる命令

#### 【0340】

なお、これら7種類の命令は、一例であって、命令グループDispCmds(i)に置かれる命令は、この7種類に限定されるものではない。表示制御命令をさらに追加してもよい。

#### 【0341】

(2) および(3)の、グラフィクスオブジェクト200の表示の開始および終了命令は、所謂フェードイン/フェードアウトを行うための命令であって、それぞれ命令fade\_in(fade\_in\_time)、命令fade\_out(fade\_out\_time)の形で用いられる。

#### 【0342】

フェードインは、命令fade\_in(fade\_in\_time)で指示され、グラフィクスオブジェクト200が非表示の状態から徐々に表示の状態に変化していく。アルファブレンディングの不透明度 $\alpha$ の値を、時刻fade\_in\_timeに基づき徐々に増加させていくことで、フェードインが実現できる。命令execution\_time(start\_time)の後に命令fade\_in(fade\_in\_time)が置かれており、命令execution\_time(start\_time)で指定される時刻start\_timeの直後から当該グラフィクスオブジェクト200が徐々に透明から不透明になっていき、引数の時刻fade\_in\_timeで指定された時間が経過すると、全てのパレットインデックスの不透明度 $\alpha$ の値が共通パレットテーブルで指定した値に設定される。

#### 【0343】

なお、命令fade\_in(fade\_in\_time)において、時刻fade\_in\_timeが0に設定されていると、当該グラフィクスオブジェクト200がパレットテーブルに指定される色および不透明度 $\alpha$ で即座に表示される。

#### 【0344】

フェードアウトは、フェードインの逆の処理であって、命令fade\_out(fade\_out\_time)で指示され、グラフィクスオブジェクト200が表示の状態から徐々に非表示の状態に変化していく。アルファブレンディングの不透明度 $\alpha$ の値を、時刻fade\_out\_timeに基づき



徐々に減少させていくことで、フェードアウトが実現できる。命令`execution_time(start_time)`の後に命令`fade_out(fade_out_time)`が置かれていると、命令`execution_time(start_time)`で指定される時刻`start_time`の直後から当該グラフィクスオブジェクト200が徐々に不透明から透明になっていき、引数の時刻`fade_out_time`で指定された時間が経過すると、全てのパレットインデックスの不透明度 $\alpha$ の値が0とされ、当該グラフィクスオブジェクト200が完全に透明になり、見えなくなる。

【0345】

なお、命令`fade_out(fade_out_time)`において、時刻`fade_out_time`が0に設定されていると、当該グラフィクスオブジェクト200の表示が即座に停止される。

【0346】

フェードインおよびフェードアウトにおいて、不透明度 $\alpha$ の値は、時間の経過と共にできるだけ滑らかに変化させると、より自然なフェードインおよびフェードアウトの効果が得られ、好ましい。これは、この例に限られず、例えば、フェードインにおいて、不透明度 $\alpha$ の値は、時刻`fade_in_time`で示される時間が経過した後に、パレットテーブルで指定した値になっていればよく、どの程度の分解能や階調で不透明度 $\alpha$ を変化させるかについては、命令によって指定していない。実際には、不透明度 $\alpha$ の変化の分解能や階調は、実装に依存する。

【0347】

なお、ここでは可読性を上げるために、命令を「`fade_in()`」、「`fade_out()`」のようにテキストで表記して説明したが、実際には、これらの命令`fade_in()`、`fade_out()`は、その引数と共に所定のバイナリ値に変換されて`DispCmds(i)`に記述される。これは、以降に説明される他の命令についても同様である。

【0348】

(4)の、使用中のパレットテーブルの色や不透明度 $\alpha$ を変更する命令は、パレット情報を変更する。この命令は、命令`change_palette(index, newR, newG, newB, newAlpha)`の形で用いられる。字幕プレーン11やグラフィクスプレーン12に同時に表示されるPNG画像は、図43に示したシンタックスで定義される、上述の図19のような共通のパレットテーブルを参照している。通常は、`GlobalPaletteTable()`で定義されたパレット情報がそのまま共通パレットテーブルとして使用される。この命令`change_palette(index, newR, newG, newB, newAlpha)`を用いることで、この共通のパレット情報を変更することができる。

【0349】

すなわち、命令`change_palette(index, newR, newG, newB, newAlpha)`に引数として記述される値`index`、`newR`、`newG`、`newB`および`newAlpha`により、パレット22において、パレット番号`index`で示されるカラーインデックス値の三原色の値R、GおよびBが値`newR`、`newG`および`newB`にそれぞれ変更され、不透明度 $\alpha$ の値が値`newAlpha`に変更される。

【0350】

(5)の、グラフィクスオブジェクトのプレーン上での表示位置やサイズを設定する命令は、命令`set_display_box(x1,y1,x2,y2)`の形で用いられ、プレーン上の座標 $(x1,y1)$ および $(x2,y2)$ で定められる矩形領域 $(x1,y1)(x2,y2)$ に、当該グラフィクスオブジェクト200を配置する。また、(6)の、グラフィクスオブジェクトの中の表示範囲を設定する命令は、命令`set_clipping_box(a1,b1,a2,b2)`の形で用いられ、グラフィクスオブジェクト200によるPNG画像上の座標 $(a1,b1)$ および $(a2,b2)$ で定められる矩形領域 $(a1,b1)(a2,b2)$ を、実際にプレーン上に表示する。

【0351】

図45を用いて、命令`set_display_box(x1,y1,x2,y2)`および命令`set_clipping_box(a1,b1,a2,b2)`について、より具体的に説明する。なお、図45における座標は、図46に一例が示されるように、表示画面の左上隅を原点とし、水平右方向をx、垂直下方向をyとして、座標 $(x, y)$ と表す。

【0352】

10

20

30

40

50

命令`set_clipping_box(a1,b1,a2,b2)`により、図45Aに示されるように、グラフィクスオブジェクト200によるPNG画像250内の、実際に表示される矩形領域( $a1,b1$ )( $a2,b2$ )が設定される。この図45Aの例では、設定される矩形領域( $a1,b1$ )( $a2,b2$ )がPNG画像250に対して小さいものとされている。命令`set_display_box(x1,y1,x2,y2)`により、この矩形領域( $a1,b1$ )( $a2,b2$ )のプレーン上における実際の表示位置が矩形領域( $x1,y1$ )( $x2,y2$ )に設定される(図45B参照)。すなわち、画面上の矩形領域( $x1,y1$ )( $x2,y2$ )に対して、PNG画像250のうち矩形領域( $a1,b1$ )( $a2,b2$ )の部分が表示される。

#### 【0353】

ここで、矩形領域( $a1,b1$ )( $a2,b2$ )が実際に表示される矩形領域( $x1,y1$ )( $x2,y2$ )よりも大きい場合は、矩形領域( $a1,b1$ )( $a2,b2$ )のうち矩形領域( $x1,y1$ )( $x2,y2$ )の範囲のPNG画像だけが表示される。一方、矩形領域( $a1,b1$ )( $a2,b2$ )の大きさが実際に表示される矩形領域( $x1,y1$ )( $x2,y2$ )よりも小さい場合は、矩形領域( $x1,y1$ )( $x2,y2$ )内の矩形領域( $a1,b1$ )( $a2,b2$ )の外側は、透明色として扱うとよい。

#### 【0354】

上述した表示制御命令を、命令`execution_time(start_time)`を複数用いて並べることで、時間経過に従って変化する字幕や同期型グラフィクスを表示することができる。例えば、図42を用いて説明したグラフィクスオブジェクト200において、ブロック`GOBJCommandTable()`内に、複数の命令グループ`DispCmds(i)`を記述する。そして、命令グループ`DispCmds(i)`のそれぞれは、命令`execution_time(start_time)`における時刻`start_time`を所望に異ならせて記述し、時刻`start_time`で示される時刻に開始する表示制御命令を記述する。

#### 【0355】

図47は、命令グループ`DispCmds(i)`の記述とグラフィクスオブジェクト200の表示変化の一例を示す。図47において、横軸は時間の経過を表し、縦軸はグラフィクスオブジェクト200の不透明度を表す。不透明度は、図47で上に向かうほど大きくなる。`pts`で示される時刻が原点とされる。

#### 【0356】

最初の命令グループ`DispCmds(0)`により、プレーン上の表示領域が命令`set_display_box(800,800,1300,900)`として、グラフィクスオブジェクト200のPNG画像における表示領域が命令`set_clipping_box(0,0,500,100)`として設定し、命令`fade_in(2 sec)`により、2秒かけてフェードインする処理を、時刻[0]から開始することが指示される。次の命令グループ`DispCmds(1)`は、命令`change_palette(index, newR, newG, newB, Alpha)`がカラーインデックス値[1]、[2] [3] および [4] についてそれぞれ記述され、時刻[800]から、パレットテーブル22においてカラーインデックス値[1]、[2]、[3] および [4] で参照される色および不透明度 $\alpha$ を変更する旨が指示される。そして、次の命令グループ`DispCmds(2)`により、表示されているグラフィクスオブジェクト200を、時刻[2000]から2秒かけてフェードアウトすることが指示される。

#### 【0357】

この図47に示されるように、命令グループ`DispCmds(0)`、`DispCmds(1)`および`DispCmds(2)`を並べることで、時間経過に従って表示が変化する例えば字幕を実現することができる。すなわち、命令グループ`DispCmds(0)`、`DispCmds(1)`および`DispCmds(2)`を適当に用いることで、字幕やボタン画像のアニメーション表示が可能とされる。

#### 【0358】

図48は、字幕表示が徐々に現れるフェードインの例を示す。図48A～図48Dと、徐々に字幕表示が現れるように表示制御される。このような単純なフェードインは、上述した図47の命令グループ`DispCmds(0)`と同様の命令で実現することができる。

#### 【0359】

図49は、字幕のPNG画像260がプレーン内を移動する例を示す。これは、命令`set_display_box(x1,y1,x2,y2)`を複数、用いることで実現できる。例えば、最初の命令グループ`DispCmds(0)`において、命令`execution_time(start_time)`により開始時刻が設定され

、命令set\_clipping\_box(a1,b1,a2,b2)により図49Aに一例が示されるようにPNG画像260の表示領域を設定すると共に、命令set\_display\_box(x1,y1,x2,y2)により、PNG画像260のプレーン内での初期の表示領域を設定する。

#### 【0360】

そして、次の命令グループDispCmds(1)において、命令execution\_time(start\_time)により命令グループDispCmds(0)から所定時間経過した時刻が開始時刻として指定されると共に、命令set\_display\_box(x1',y1',x2',y2')により、プレーン内の移動先の表示領域が設定される。同様に、次の命令グループDispCmds(2)において、命令execution\_time(start\_time)により命令グループDispCmds(1)から所定時間経過した時刻が開始時刻として指定されると共に、命令set\_display\_box(x1'',y1'',x2'',y2'')により、プレーン内の移動先 10  
の表示領域が設定される。

#### 【0361】

このようにすることで、図49Bに一例が示されるように、字幕のPNG画像260がプレーン上の矩形領域(x1,y1)(x2,y2)、矩形領域(x1',y1')(x2',y2')、矩形領域(x1'',y1'')(x2'',y2'')と移動するようにできる。

#### 【0362】

図50は、字幕のPNG画像261内の表示領域262が移動し、字幕表示をスクロールさせる例を示す。これは、命令set\_clipping\_box(a1,b1,a2,b2)を複数、用いることで実現できる。例えば、最初の命令グループDispCmds(0)において、命令execution\_time(start\_time)により開始時刻が設定され、命令set\_clipping\_box(a1,b1,a2,b2)により、図5 20  
0Aに一例が示されるように、PNG画像260において初期に表示される矩形領域262を設定すると共に、命令set\_display\_box(x1,y1,x2,y2)により、PNG画像260がプレーン内で表示される矩形領域を設定する。

#### 【0363】

そして、次の命令グループDispCmds(1)において、命令execution\_time(start\_time)により命令グループDispCmds(0)から所定時間経過した時刻が開始時刻として指定されると共に、命令set\_clipping\_box(a1',b1',a2',b2')により、PNG画像260内の移動先の表示領域が設定される。同様に、次の命令グループDispCmds(2)において、命令execution\_time(start\_time)により命令グループDispCmds(1)から所定時間経過した時刻が開始時刻として指定されると共に、命令set\_clipping\_box(a1'',b1'',a2'',b2'')により、PNG 30  
画像260内の移動先の矩形領域が設定される。

#### 【0364】

このようにすることで、図50Bに一例が示されるように、プレーン上の矩形領域(x1,y1)(x2,y2)内を、字幕のPNG画像261内の一部の矩形領域が矩形領域(a1,b1)(a2,b2)、矩形領域(a1',b1')(a2',b2')、矩形領域(a1'',b1'')(a2'',b2'')と移動するようにされ、字幕表示のスクロールを実現できる。

#### 【0365】

図51は、PNG画像265の一部を表示する枠を設定し、PNG画像265上でこの枠を移動させつつ、プレーン上での表示位置も移動させる例を示す。これは、命令set\_display\_box(x1,y1,x2,y2)と、命令set\_clipping\_box(a1,b1,a2,b2)とを複数、同時に用い 40  
ることで実現できる。例えば、最初の命令グループDispCmds(0)において、命令execution\_time(start\_time)により開始時刻が設定され、命令set\_display\_box(x1,y1,x2,y2)と命令set\_clipping\_box(a1,b1,a2,b2)とにより、枠266Aが設定される(図51A参照)。

#### 【0366】

例えば、命令set\_display\_box(x1,y1,x2,y2)によりプレーン上で表示される矩形領域(x1,y1)(x2,y2)が設定されると共に、命令set\_clipping\_box(a1,b1,a2,b2)によりPNG画像265において表示される矩形領域(a1,b1)(a2,b2)が設定される。これら矩形領域(x1,y1)(x2,y2)および矩形領域(a1,b1)(a2,b2)により、枠266Aが設定される。

#### 【0367】

そして、次の命令グループDispCmds(1)において、命令execution\_time(start\_time)により命令グループDispCmds(0)から所定時間経過した時刻が開始時刻として指定され、命令set\_display\_box(x1',y1',x2',y2')によるプレーン内の矩形領域(x1',y1')(x2',y2')と、命令set\_clipping\_box(a1',b1',a2',b2')によるPNG画像265上の矩形領域(a1',b1')(a2',b2')により、枠266Aに対する移動先の枠266Bが設定される。同様に、次の命令グループDispCmds(2)において、命令execution\_time(start\_time)により命令グループDispCmds(1)から所定時間経過した時刻が開始時刻として指定され、命令set\_display\_box(x1",y1",x2",y2")によるプレーン内の矩形領域(x1",y1")(x2",y2")と、命令set\_clipping\_box(a1",b1",a2",b2")によるPNG画像265上の矩形領域(a1",b1")(a2",b2")により、枠266Bに対する移動先の枠266Cが設定される。

10

【0368】

こうすることで、図51Bに一例が示されるように、字幕のPNG画像265内の一部の矩形領域が移動しつつ、当該矩形領域がプレーン内を領域265A、265B、265Cと移動するような表示が可能となる。

【0369】

このように、この発明の実施の第1の形態では、グラフィクスオブジェクト200の表示制御を、命令execution\_time(start\_time)により各表示制御命令をグループ化した命令グループDispCmds(i)を用いて行っているため、字幕プレーン11およびグラフィクスプレーン12における多彩な表示を、容易に実現することができる。

【0370】

20

2-13. 効果音について

この発明の実施の第1の形態では、さらに、グラフィクスオブジェクト200の表示制御に同期して音声を出力することができる。上述した、命令グループDispCmds(i)に置かれる、命令execution\_time(start\_time)以外の表示制御命令(2)～(8)のうち、(7)の効果音の再生を行う命令と、(8)の画像データに効果音を割り当てる命令により、音声出力が定義される。また、音声データには、それぞれ識別子sound\_idが付与され、互いが識別される。

【0371】

なお、ここでいう効果音とは、動画プレーン上に表示される動画や静止画に同期して再生される音声(例えば、映画などにおいて、映画映像と対になって収録されている音声)とは別途に用意された音声データが、字幕やボタンの表示制御に同期または誘引されて再生される音声のことをいう。

30

【0372】

背景技術でも説明したが、ユーザは、例えば、リモートコントロールコマンド(リモコン)に設けられた、上下左右の方向をそれぞれ指示するようにされたキー(キー(方向キー)などを操作して、メニュー画面に表示されたボタンの一つを選択状態とし、そこで決定キーを操作することで、選択されたボタンに割り当てられた動作を実行状態とする。また、一つのボタンには、「通常(非選択)状態」、「選択状態」および「実行状態」という3種類の状態が存在する。一つのボタンの3種類の状態に対してそれぞれ異なるグラフィクスオブジェクトを対応させることができる。これにより、ボタンの3種類の状態毎に、ボタン画像(形状や色など)を変更することが可能である。こうすることで、ユーザがボタンの状態を区別し易くなり、好ましい。

40

【0373】

さらに、ボタンの3種類の状態毎に、異なる効果音を対応付けることができる。例えば、ユーザが、ある一つのボタンを選択状態にした際に発せられる効果音を「選択状態」のボタン画像に関連付け、また、当該ボタンを実行状態にした際に発せられるクリック音のような効果音を「実行状態」のボタン画像に関連付ける。

【0374】

「選択状態」のボタン画像に関連付けられた効果音は、当該効果音を再生する音声データの識別子sound\_idで参照され、ユーザが当該ボタンを選択状態にした際に、その識別子

50

sound\_idで参照される音声データがプレーヤのメモリから読み出されて、効果音として再生される。同様に、「実行状態」のボタン画像に関連付けられた効果音は、当該効果音を再生する音声データの識別子sound\_idで参照され、ユーザがボタンの選択状態時に決定キーを操作するなどして、当該ボタンを実行状態にした際に、「実行状態」のボタン画像に関連付けられた効果音の識別子sound\_idにより参照される音声データが、プレーヤのメモリから読み出されて再生される。

#### 【0375】

上述した、命令グループDispCmds(i)に置かれる表示制御命令のうち(7)の、効果音の再生を行う命令は、命令play\_sound(sound\_id)の形で用いられ、識別子sound\_idで指定された音声データの再生を指示する。この命令を命令グループDispCmds(i)に含めることにより、命令execution\_time(start\_time)で指定された時刻start\_timeに、識別子sound\_idで指定された音声データが再生される。

#### 【0376】

一例として、命令play\_sound(sound\_id)を命令fade\_in(fade\_in\_time)および命令fade\_out(fade\_in\_time)と同時に使用することで、字幕の表示および／または消去に連動して音声データを再生し、効果音を発生させることが可能となる。図52は、上述した図47に対して命令play\_sound(sound\_id)を追加した例を示す。図52の例では、最初の命令グループDispCmds(0)において、開始時刻[0]で2秒かけて表示がフェードインすると共に、命令play\_sound(1)により識別子sound\_idが[1]で特定される音声データが再生される。次いで、命令グループDispCmds(1)において、時刻[800]で表示色が変更されると共に、命令play\_sound(2)により識別子sound\_idが[2]で特定される音声データが再生される。そして、命令グループDispCmds(2)において、時刻[2000]から1秒かけて表示がフェードアウトすると共に、命令play\_sound(1)により識別子sound\_idが[1]で特定される音声データが再び再生される。

#### 【0377】

(8)の、PNGデータに効果音を割り当てる命令は、命令set\_sound(PNG\_image\_id, sound\_id)の形で用いられ、識別子PNG\_image\_idで指定されるPNGデータに対して、識別子sound\_idで指定される音声データを割り当てる。この命令set\_sound(PNG\_image\_id, sound\_id)により、識別子PNG\_image\_idで指定されるPNGデータが表示される際に、識別子sound\_idで指定される音声データが再生される。PNGデータの識別子PNG\_image\_idは、例えば、上述した図42における、ブロックPNGImageRegion()中のPNG\_image(i)に対して指定されるループカウンタiと同じ値を用いることができる。

#### 【0378】

この命令set\_sound(PNG\_image\_id, sound\_id)は、主に、選択状態および実行状態のボタンのPNGデータに用いることが想定されている。こうすることで、ボタンが通常状態から選択状態へ遷移する際や、選択状態から実行状態に遷移する際に、それぞれの状態を表すPNGデータに割り当てられた音声データを効果音として発生させることが可能となる。勿論、この例に限らず、ボタン以外のPNGデータに対してこの命令set\_sound(PNG\_image\_id, sound\_id)を用いてもよい。

#### 【0379】

図53は、ボタン画像に音声データを割り当てたグラフィクスオブジェクト200の一例のデータ構造を示す。PNGデータ領域203に、通常状態、選択状態および実行状態それぞれにおけるボタンのPNGデータ203A、203Bおよび203Cが格納される。この図53の例では、表示制御命令ではPNGデータに対する座標指定と、音声データの割り当てのみを行い、PNGデータの表示開始時刻や、表示開始時のボタンの初期状態は、外部のスクリプトプログラムにより制御するようにしている。そのため、表示制御命令は、便宜上、実行時刻[0]の命令として記述する。

#### 【0380】

図53に例示されるグラフィクスオブジェクト200では、表示制御命令テーブル202において、命令execution\_time(0)に基づき時刻[0]で実行される、命令グループDis

pGmids(0)だけが存在している。識別子PNG\_image\_idは、[0]から始まるので、識別子PNG\_image\_idが[0]で通常状態時のPNGデータ203Aを、識別子PNG\_image\_idが[1]で選択状態時のPNGデータ203Bを、識別子PNG\_image\_idが[2]で実行状態時のPNGデータ203Cをそれぞれ表している。

#### 【0381】

命令set\_sound(1,10)に基づき、識別子PNG\_image\_idが[1]である選択状態のボタンを表示するPNGデータ203Bが表示されたときには、識別子sound\_idが[10]で指定される音声データが効果音として再生される。同様に、命令set\_sound(2,11)に基づき、識別子PNG\_image\_idが[2]である実行状態のボタンを表示するPNGデータ203Cが表示されたときには、識別子sound\_idが[11]で指定される音声データが効果音として再生されることになる。

#### 【0382】

上述の図34では省略されているが、プレーヤは、予め1または複数種類の音声データを、プレーヤ内部のメモリなどに保持することができる。例えば、プレーヤは、プレーヤ内部に設けられた不揮発性メモリなどに、出荷時に予め所定の音声データを記憶しておくようにできる。

#### 【0383】

また、これに限らず、グラフィックオブジェクト200や動画データによるコンテンツが記録されたディスクに、効果音として用いるための音声データを記録しておき、当該ディスクの再生時にこの音声データを読み込んでおくようにしてもよい。ディスクへの音声データの格納方法としては、音声データが格納されたファイルを別に用意し、ディスク再生開始時にこのファイルを先に読み込んでおく、プレーヤ内のメモリに置いておく方法がある。

#### 【0384】

あるいは、グラフィックスオブジェクト200と同様に音声データを格納したPESパケットを作り、それをTSパケットに分割してクリップAVストリームに多重化する方法も考えられる。

#### 【0385】

さらには、図37には示していないが、図37のグラフィックスオブジェクト200中のグラフィックスオブジェクトヘッダ201内、あるいは、PNG画像データ領域203の直後に、グラフィックスオブジェクト200に含まれるPNG画像に対応する音声データを格納する方法も考えられる。

#### 【0386】

何れの方法であっても、音声データを予めディスクからプレーヤのメモリに読み込んで蓄積しておくことができるので、PNG画像で作られたボタンが選択状態や実行状態になると同時に効果音を発生させることが可能となる。音声データには、それぞれユニークな識別子sound\_idが割り当てられており、音声データを一意に特定することができるようにされている。

#### 【0387】

2-14. 音声データのディスクへの格納方法について

この、音声データのディスクへの格納方法について、より詳細に説明する。以下では、音声データのディスクへの格納方法として、次の3つの方法について説明する。

(1) 音声データをグラフィックスオブジェクト200に多重化する方法

(2) 音声データをクリップAVストリームに多重化せず、且つ、識別子sound\_id毎の音声データファイルとする方法

(3) 音声データをクリップAVストリームに多重化せず、且つ、複数の音声データをまとめて一つのデータファイルとする方法

#### 【0388】

2-14 a. 第1の格納方法

先ず、(1)の、音声データをグラフィックスオブジェクト200に多重化する方法（以

下、第1の格納方法と呼ぶ)について、図54を用いて説明する。図54Aおよび図54Bは、音声データをグラフィクスオブジェクト200に付加してクリップAVストリームに多重化する例を示す。

#### 【0389】

図54Aは、グラフィクスオブジェクト200中のPNGデータ領域203の後ろに音声データ領域204を設けた例である。音声データ領域204には、複数の音声データ204A、204B、・・・、204nを格納することができる。これら音声データ204A、204B、・・・、204nを、同一のグラフィクスオブジェクト200に格納されるPNGデータ203A、203B、・・・、203nとそれぞれ関連性のあるものとする、PNG画像および音声データを対応付けての扱いが容易になる。

10

#### 【0390】

音声データ204A、204B、・・・、204nは、例えばA I F F (Audio Interchange File Format)形式やW A V E ファイルといった、圧縮符号化されていないデータでもよいし、M P 3 (Moving Pictures Experts Group 1 Audio Layer 3)やA A C (Advanced Audio Coding)、A T R A C (Adaptive Transform Acoustic Coding)といった、圧縮符号化されたデータでもよい。圧縮符号化された音声データを格納する場合には、プレーヤ側が音声データの圧縮符号化方式に対応したオーディオデコーダを持っている必要がある。

#### 【0391】

図54Bは、ボタンを構成するグラフィクスオブジェクト200に対して、ボタンのそれぞれの状態に対応した音声データを格納した例である。この例では、例えばボタンの選択状態時に再生する音声データ204A-2と、実行状態時に再生する音声データ204B-2とが、音声データ領域203に格納される。一方、PNGデータ領域203には、通常状態、選択状態および実行状態それぞれのボタン表示に用いるPNGデータ203A-2、203B-2および203C-2が格納される。

20

#### 【0392】

この場合、例えば、ボタン画像が格納されるPNGデータ領域203に続く音声データ領域204には、ボタンが選択状態時に遷移したときに再生する音声データ204A-2と、実行状態時に遷移したときに再生する音声データ204B-2とが格納されるように、予め決めておく。そして、ボタン画像のPNGデータが表示されると同時に、当該ボタンの状態に対応した音声データが再生されるようにする。このプレーヤで再生される効果音は、主にボタンクリック音として使われることを想定しているため、このようなルールでも、十分に、主要な目的を達成することができる。

30

#### 【0393】

図55は、図39を用いて説明したグラフィクスオブジェクトデコーダモデル240に対して音声データの処理系を追加した、グラフィクスオブジェクトデコーダモデル240'の一例の構成を示す機能ブロック図である。なお、図55において、図39と共通する部分には同一の符号を付し、詳細な説明を省略する。

#### 【0394】

端子202からM P E G T Sとして入力された、グラフィクスオブジェクト200に音声データが格納されたクリップAVストリームがP I Dフィルタ110に供給され、P I Dに基づき動画像データ、音声データおよびグラフィクスオブジェクト200をそれぞれ抜き出す。抜き出されたデータのうち、グラフィクスオブジェクト200は、バッファT B n 1 1 1 Bを介してG O B J パーサ224に供給される。G O B J パーサ224では、供給されたグラフィクスオブジェクト200のグラフィクスオブジェクトヘッダ201を読み取り、パレットデータを抽出すると共に、表示制御命令テーブル202、PNGデータ領域203および音声データ領域204を分離する。

40

#### 【0395】

パレットデータおよび表示制御命令テーブル202は、コマンドプロセッサ/グラフィックレンダラ225に供給される。さらに、音声データ領域204の音声データ204A、204B、・・・、204nがコマンドプロセッサ/グラフィックレンダラ225に供

50



給され、図示されないバッファにそれぞれ溜め込まれる。

#### 【0396】

コマンドプロセッサ／グラフィックレンダラ225は、GOBJパーサ224から供給された表示制御命令テーブル202に記述される表示制御命令に従い、バッファから音声データを読み出し、出力する。グラフィクスオブジェクト200に格納された音声データ204A、204B、・・・、204nが圧縮符号化されている場合には、コマンドプロセッサ／グラフィックレンダラ225において復号化されて出力される。

#### 【0397】

コマンドプロセッサ／グラフィックレンダラ225から出力された音声データは、オーディオミキサ231に供給され、プレゼンテーションプロセッサ141に対して出力される。オーディオミキサ231に他の音声データも入力されている場合には、これらの音声データが所定の割合で混合されて出力される。

#### 【0398】

図56は、上述の図54Aおよび図54Bのようにして音声データを加えた場合の、グラフィクスオブジェクト200の一例の構造を示すシンタックスを示す。このように、上述した図42の構造に対し、ブロックGraphicsObjectHeader()のフィールドnumber\_of\_PNG\_imagesの次に、フィールドnumber\_of\_sound\_dataが配置され、ブロックSoundDataRegion()に格納されている音声データの数データ長が8ビットの0以上の整数で示される。フィールドstart\_address\_of\_sound\_data(i)は、データ長が32ビットの0以上の整数で、i番目の音声データsound\_data(i)が開始される位置を、このブロックGraphicsObject()の先頭からの相対バイト数で表現する。また、ブロックPNGImageRegion()の後にpadding\_wordが挿入され、続けてブロックSoundDataRegion()が配置される。このブロックSoundDataRegion()に、音声データsound\_data(i)として実際の音声データが格納される。

#### 【0399】

##### 2-14b. 第2の格納方法

次に、(2)の、音声データをクリップAVストリームに多重化せず、且つ、識別子sound\_id毎の音声データファイルとする方法（以下、第2の格納方法と呼ぶ）について説明する。例えば、図57に一例が示されるように、ディレクトリBD AVの下に音声データを格納するディレクトリSOUNDを設ける。このディレクトリSOUNDには、音声データであるPCM波形データを格納する、例えばA I F F形式の音声データファイル「sound1.aiiff」を置く。ディレクトリSOUNDに格納された音声データファイルは、例えば当該ディスクのプレーヤへの最初のローディング時にまとめて読み込まれ、プレーヤの内部メモリに格納される。

#### 【0400】

各音声データには、音声データそれぞれを識別可能な識別子sound\_idが予め割り当てられ、スクリプトなどから識別子sound\_idにより、所望の音声データがメモリから呼び出されるモデルが構成される。

#### 【0401】

この場合、例えば、グラフィクスオブジェクト200に対して図54Cに一例が示されるように、サウンドid領域205を設け、このサウンドid領域205に対してサウンドidデータ205A、205Bを格納する。この図54Cの例では、PNGデータ領域203にボタンの通常状態時、選択状態時および実行状態時にそれぞれ対応したPNGデータ203A、203Bおよび203Cが格納され、サウンドidデータ205Aおよび205Bは、PNGデータ203Bおよび203Cにそれぞれ対応する識別子sound\_idである。PNGデータ203Bの表示の際に、プレーヤ内部のメモリに予め格納された、サウンドidデータ205Aが示す識別子sound\_idが対応する音声データを再生するように指示される。

#### 【0402】

また例えば、図53を用いて既に説明したように、表示制御命令テーブル202の記述によって、PNGデータと音声データとを関連付けるようにしてもよい。

## 【0403】

識別子sound\_idにより音声データを指定するため、図44を用いて既に説明した表示制御コマンドなどにより、グラフィクス表示とは関係の無い任意の時刻に、音声データによる効果音などを発生させることができる。

## 【0404】

一方、この方法では、識別子sound\_idで音声データをメモリから呼び出すため、効果音の種類は、第1に、識別子sound\_idの数で制限される。また、第2に、使える効果音の種類が、上述したように、プレーヤの内部メモリの容量により制限される。

## 【0405】

図58を用いて、より具体的に説明する。ディスク400のローディング時に、プレーヤによりディスクに対してインシヤルアクセスがなされ、ディレクトリBD A Vの下に配置されるディレクトリS O U N Dから、音声データがまとめて読み込まれる。読み込まれた音声データ（PCMデータ）は、プレーヤの内部メモリ410に格納される。このとき、音声データのそれぞれに対して固有の識別子sound\_idが割り当てられる。ディスク400に記録されている音声データに対して予め識別子sound\_idを付加しておいてもよい。

## 【0406】

この例では、16個の音声データがディスク400から読み込まれ、識別子sound\_id=1~16がそれぞれの音声データに割り当てられる。また、音声データそれぞれのデータサイズが取得される。この図58の例では、識別子sound\_id=1~16の音声データがそれぞれd1バイト、d2バイト、・・・、d16バイトのデータサイズを有しているものとする。

## 【0407】

例えば、ボタン420A、420Bおよび420Cが表示されるメニュー画面420における、ボタン420Cに対する操作に基づき、ボタン420Cに対して指定された識別子sound\_idにより対応する音声データがメモリ410から読み出される。この図58の例では、ボタン420Cの実行状態に対して識別子sound\_id=1で示される音声データが指定されている。メモリ410から読み出されたこの音声データは、所定に処理されてバッファ450Bに一旦溜め込まれる。そして、オーディオミキサ231に供給され、例えば本編の動画データに付随する音声データと混合されて、音声として出力される。

## 【0408】

なお、バッファ450Aには、例えば本編の動画データに付随する音声データが一旦、溜め込まれる。例えば、バッファ450Aおよび450Bの間で、溜め込まれた音声データを読み出すタイミングを調整することで、バッファ450Bに溜め込まれたボタン420Cの操作に対応した効果音がバッファ450Aに溜め込まれた音声データの適当なタイミングで出力される。また、この例では、識別子sound\_id=0で、音声データの再生無しが指定される。

## 【0409】

このようなモデルにおいて、ディスク400から読み込み可能な音声データの合計容量がメモリ410の容量に基づき制限される。また、それぞれの音声データの容量がバッファ450Bの容量に基づき制限される。すなわち、メモリ410の容量を容量M（バイト）、バッファ450Bの容量を容量Dmax（バイト）とした場合、次の2つの条件を満たす必要がある。

## 【0410】

(1) メモリ410に格納された音声データのそれぞれの容量d<sub>i</sub>がバッファ450Bの容量Dmaxより小さい。

(2) メモリ410に格納された音声データの総容量（d<sub>1</sub>+d<sub>2</sub>+・・・+d<sub>n</sub>）がメモリ410の容量Mより小さい。

## 【0411】

換言すれば、この(1)および(2)の条件をプレーヤ側とディスク制作側とに対してルール化し、規格として定めておくことで、効果音などの音声データの再生に関して再生互換性を確保することができる。

## 【0412】

なお、ここでは、音声データをクリップAVストリームに多重化しない場合（第2の格納方法）に、音声データをディスクのプレーヤへの最初のローディング時にまとめて読み込むように説明したが、これはこの例に限定されない。一例として、音声データを複数回に分けて読み出すようにできる。例えば、シナリオの進行に応じてある区切となる箇所、次の区切りまでに用いるだけの音声データをまとめて読み出し、メモリに記憶させる。このとき、それまでメモリに記憶されていた音声データを消去する。こうすることで、1つのシナリオの再生において、メモリの容量以上の音声データを扱うことが可能とされる。

## 【0413】

10

音声データは、ディスクの所定領域にまとめて記録しておいてもよいし、ディスクの複数領域に分けて記録しておいてもよい。ディスクの複数領域に分けて記録する場合には、例えば、シナリオの進行に応じて、シナリオの区切りに対応したディスク上の位置に、次の区切りまでに用いられる音声データをまとめて記録することが考えられる。

## 【0414】

なお、図54Aおよび図54Bを用いて説明した、音声データをクリップAVストリームに多重化する方法（第1の格納方法）では、音声データの種類の制約が無く、1画像毎に異なる音声データを指定できる利点がある。すなわち、音声データは、必要に応じてクリップAVストリームにより供給されるため、その都度異なる音声データを用いることができる。また、第1の格納方法では、音声データが画像データと同時にクリップAVストリームから読み込まれるため、読み込みモデルが簡単になる、音声データのファイル数の制限やファイルサイズの制限は、ディスク容量以外の制約を受けない、などの利点がある。

20

## 【0415】

一方、第1の格納方法では、異なるグラフィクスオブジェクトで同一の音声データを扱うような場合、それぞれのグラフィックスオブジェクトが同一の音声データをそれぞれ持たなければいけないので、データが冗長になってしまう。また、グラフィックスオブジェクトから音声データを抽出する必要があるため、クリップAVストリームのデマルチプレクス処理の後に、音声データを分離する処理が必要となる。

## 【0416】

30

2-14c. 第3の格納方法

次に、(3)の、音声データをクリップAVストリームに多重化せず、且つ、複数の音声データをまとめて一つのデータファイルとする方法（以下、第3の格納方法とする）について説明する。複数の音声データは、例えばファイル「HdmvSound.bdmv」にまとめて格納される。ファイル「HdmvSound.bdmv」は、図59に一例が示されるように、上述の図4に示した「Blu-ray Disc Rewritable Format Ver1.0 part3」で規定された、記録媒体に記録されるファイルの管理構造に対し、ディレクトリBD-AVの直下に置かれる。ディスクがプレーヤに装填されると、このファイル「HdmvSound.bdmv」がイニシャルアクセス時に読み出され、例えばそのディスクに主コンテンツとして収録されているクリップAVストリーム（例えば映画など）に先立って、プレーヤが有するメモリに記憶されプリロードされる。

40

## 【0417】

なお、図59において、ディレクトリSTREAMは、例えば映画など、当該ディスクの主コンテンツのクリップAVストリームが格納される。

## 【0418】

図60は、ファイル「HdmvSound.bdmv」の一例のシンタクスを示す。フィールドlengthは、32ビットのデータ長を有し、フィールドlengthの直後のフィールドからこのファイル「HdmvSound.bdmv」の最後まで長さがバイトで示される。フィールドdata\_block\_start\_addressは、32ビットのデータ長を有し、ファイル「HdmvSound.bdmv」の中において音声データがまとめて配置されるブロックdata\_blockの、先頭バイトの開始アドレスを示

50

す。フィールド `number_of_sound_entries` は、8 ビットのデータ長を有し、ファイル「`HdmvSound.bdmv`」の中に格納されている音声データ（すなわち、効果音データ）の数を示す。

#### 【0419】

`for` ループの引数によって、一つの音声データを参照する識別子 `sound_id` が自動的に与えられる。この `for` ループ中に、引数に対応する識別子 `sound_id` で参照される音声データの情報が記述される。

#### 【0420】

ブロック `attributes` は、当該音声データの属性を示す。フィールド `channel_assignment` は、当該音声データのチャンネルアサインメントであり、例えば、モノラルや2チャンネルステレオという属性である。フィールド `sampling_frequency` は、当該音声データのサンプリング周波数であり、例えば 48 kHz である。フィールド `bits_per_sample` は、当該音声データの量子化ビット数であり、例えば 16 ビットである。

#### 【0421】

フィールド `sound_data_start_address` は、32 ビットのデータ長を有し、当該 `for` ループのフィールド `sound_id` で参照される音声データの先頭バイトがブロック `data_block` の中で開始するアドレスを示す。フィールド `sound_data_length` は、32 ビットのデータ長を有し、当該音声データのバイト長である。この `for` ループの終了後、パディングワードを挟んで、ブロック `data_block` が配置される。ブロック `data_block` は、音声データが実際に格納される領域である。

#### 【0422】

図 6 1 は、上述のシンタクスによるファイル「`HdmvSound.bdmv`」の構成を示す。ファイルの先頭からフィールド `number_of_sound_entries` までをヘッダ部 600 とする。ヘッダ部 600 に続けて、`for` ループ部 601 が配置され、次に音声データ部 602 が配置される。

#### 【0423】

音声データ部 602 には、識別子 `sound_id=1` ~ 識別子 `sound_id=n` で参照される音声データが連続的に配置される。例えば、ある音声データの末尾と次の音声データの先頭とがバイト境界で接続されて、識別子 `sound_id=1` ~ 識別子 `sound_id=n` で参照される音声データが次々に配置される。音声データ部 602 では、`for` ループ部 601 中の、引数が識別子 `sound_id` に対応するループにおけるフィールド `sound_data_start_address` により先頭アドレスが指定され、その指定アドレスから当該ループのフィールド `sound_data_length` で指定される長さまでが、当該ループの引数に対応する識別子 `sound_id` の音声データとして参照される。

#### 【0424】

図 6 2 は、この第 3 の格納方法による効果音再生の一例のアプリケーションイメージを示す。なお、図 6 2 において、上述した図 5 8 と共通する部分には同一の符号を付し、詳細な説明を省略する。ディスク 400 のローディング時のプレーヤによるディスク 400 に対するイニシャルアクセスにより、ディレクトリ `BDAV` の下に配置されるファイル「`HdmvSound.bdmv`」が予め読み込まれる（プリロード）。読み込まれたファイル「`HdmvSound.bdmv`」は、プレーヤの内部メモリ 410 に格納される。

#### 【0425】

図 6 1 を用いて既に説明したように、音声データのそれぞれは、ファイル「`HdmvSound.bdmv`」のループ部 601 でアドレス情報およびデータ長情報とに基づき指定され、対応する識別子 `sound_id` により識別される。この図 6 2 の例では、ファイル「`HdmvSound.bdmv`」は、 $m$  個の音声データを含み、識別子 `sound_id=1, 2, \dots, m` が  $m$  個の音声データに対してそれぞれ割り当てられる。それぞれの音声データの容量は、例えば識別子 `sound_id` に対応するデータ長情報および量子化ビット数 `bits_per_sample` から知ることができる。

#### 【0426】

なお、ここでは、音声データがPCMにより符号化されているとして説明したが、これはこの例に限定されない。音声データは、MPEG1レイヤ3 (mp3) 方式やAAC (Advanced Audio Coding)方式、ATRAC (Adaptive Transform Acoustic Coding)方式など所定の方式で圧縮符号化されて音声データ部602に格納されてもよい。この場合でも、各音声データは、ある音声データの末尾と次の音声データの先頭が連続するようにして、音声データ部602に格納される。

#### 【0427】

例えば、ボタン420A、420Bおよび420Cが表示されるメニュー画面420を考える。メニュー画面420において、ボタン420Cは、通常状態、選択状態および実行状態でそれぞれ異なるボタン画像420C-1、420C-2および420C-3を有するボタン画像セット430からなる。このように、ボタンの3種類の状態に応じてボタン画像を変更すると、ユーザがボタン状態を区別し易く、好ましい。また、この図62の例では、選択状態のボタン画像420C-2に対して識別子sound\_id=2の音声データが対応付けられ、実行状態のボタン画像420C-3に対して識別子sound\_id=1の音声データが対応付けられている。なお、識別子sound\_idの値が「0」のときは、ボタン画像は表示されるが、音声は出力されないものとする。

#### 【0428】

例えばユーザのリモコン操作によりボタン420Cが選択状態から実行状態に移移すると、ボタンの表示画像がボタン画像420C-2からボタン画像420C-3に変更されると共に、ボタン画像420C-3に対応付けられた識別子sound\_id=1の音声データがメモリ410から読み出され、再生される。

#### 【0429】

すなわち、ボタン画像420C-3が表示されると共に、ボタン画像420C-3に対応付けられた識別子sound\_id=1に対応するアドレス情報およびデータ長情報に基づきメモリ410がアクセスされ、識別子sound\_id=1で示される音声データがメモリ410から読み出される。メモリ410から読み出された音声データは、バッファ450Bに一旦溜め込まれる。そして、オーディオミキサ231に供給されて、例えばバッファ450Aから出力された、本編の動画像データに付随する音声データと混合されて、音声として出力される。

#### 【0430】

なお、この第3の格納方法においても、図58を用いて説明した上述の(2)の方法の場合と同様に、ディスク400から読み込み可能な音声データの合計容量や、それぞれの音声データの容量は、メモリ410の容量や、バッファ450Bの容量に基づき制限される。

#### 【0431】

この第3の格納方法に適用できるグラフィクスオブジェクトの一例の構成について説明する。この第3の格納方法では、グラフィクスプレーン12を用いたGUI (Graphical User Interface)によるメニュー画面を構成するボタン画像を、1つのストリーム構造にまとめて符号化する。以下、このボタン画像データがまとめられたストリームを、インタラクティブグラフィクスストリームと呼ぶ。

#### 【0432】

一例として、上述した図62のメニュー画面420でいうと、ボタン420A、420Bおよび420Cを表示するためのボタン画像データがまとめてインタラクティブグラフィクスストリーム700 (図63参照)として符号化される。ボタン420Cの例のように、1つのボタンが状態に応じた複数のボタン画像データ420C-1、420C-2および420C-3を有するボタン画像セット430からなる場合、ボタン画像セット430が含む複数のボタン画像データ420C-1、420C-2および420C-3が、ボタン420Aおよび420Bのボタン画像データと共に、まとめてインタラクティブグラフィクスストリーム700として符号化される。このインタラクティブグラフィクスストリーム700は、トランスポートストリームに多重化されて、ディスクに記録される

。【0433】

図63は、インタラクティブグラフィクスストリーム700の一例の構成を概略的に示す。1つのインタラクティブグラフィクスストリーム700は、図63Aに一例が示されるように、1つのインタラクティブコンポジションセグメント701と、1または複数のボタン画像データ702、702、・・・を持つ。ここで、それぞれのボタン画像データ702、702、・・・は、異なる値の識別子object\_idを有し、この識別子object\_idにより参照される。ボタン画像データ702、702、・・・は、例えばビットマップ画像データを符号化したものであり、例えば、ビットマップ画像データをランレングス符号化して作成する。

10

【0434】

図63Bに一例が示されるように、インタラクティブグラフィクスストリーム700は、同一のPIDを有する複数のトランスポートパケットのペイロードに格納され、このトランスポートパケットによりトランスポートストリームに多重化される。

【0435】

図64は、インタラクティブコンポジションセグメント701の一例の構成を示す。インタラクティブコンポジションセグメント701は、セグメントディスクリプションアトリビュート、ディスプレイタイミングアトリビュートと、ボタンレイアウトおよびコマンドとからなる。セグメントディスクリプションアトリビュートおよびディスプレイタイミングアトリビュートは、このインタラクティブコンポジションセグメント701で定義されるボタンの表示属性や表示タイミングに関する属性情報が格納される。

20

【0436】

ボタンレイアウトおよびコマンドは、GUIによるメニュー画面を構成する1以上のボタンの情報703、703、・・・を有する。ボタン情報703において、フィールドbutton\_numberは、各ボタンを識別するための識別子button\_numberが格納される。各ボタンは、この識別子button\_numberにより識別および参照される。フィールドselected\_state\_sound\_idは、「選択状態」のボタン画像に関連付けられた音声データを参照する識別子sound\_idが格納される。フィールドactivated\_state\_sound\_idは、「実行状態」のボタン画像に関連付けられた音声データを参照する識別子sound\_idが格納される。フィールドneighbour\_infoは、図65に一例が示されるように、このボタンに対して画面上で上下左右に隣接して配置されるボタンの番号Upper\_button\_number、Lower\_button\_number、Left\_button\_numberおよびRight\_button\_numberが格納される。

30

【0437】

フィールドNormal\_state\_object\_idは、このボタンが「通常状態（非選択状態）」のときに参照されるボタン画像の識別子object\_idが格納される。フィールドSelected\_state\_object\_idは、このボタンが「選択状態」のときに参照されるボタン画像の識別子object\_idが格納される。フィールドActivates\_state\_object\_idは、このボタンが「実行状態」のときに参照されるボタン画像の識別子object\_idが格納される。フィールドcommandは、このボタンが実行状態になったときに実行される、当該ボタンに割り当てられた動作情報が格納される。

40

【0438】

2-15. プレーンの別の例について

なお、上述では、図15、図16および図17を用いて説明したように、グラフィクスプレーン12がカラーシステムとしてYCbCr(4:4:4)およびRGB(4:4:4)を選択可能として説明したが、これはこの例に限定されない。例えば、図66に一例が示されるように、グラフィクスプレーン12'をボタン用のビットマップ画像を主たる表示とするように定義し、図67に一例が示されるように、字幕プレーン11と同様に、1920画素×1080ラインで各画素のサンプリング深さが8ビットとされ、カラーシステムを256色のパレットを用いた8ビットカラーマップアドレスとしてもよい。

【0439】

50

この場合、動画像プレーン10、字幕プレーン11およびグラフィクスプレーン12'を合成する構成は、図68に一例が示されるようになる。なお、図68において、上述の図17と共通する部分には同一の符号を付し、詳細な説明を省略する。

#### 【0440】

グラフィクスプレーン12'の画像データがパレット26Aに入力され、RGB(4:4:4)の画像データとして出力される。この画像データに対してアルファブレンディングによる不透明度が指定されている場合には、指定された不透明度 $\alpha 2$ ( $0 \leq \alpha 2 \leq 1$ )がパレット26Aから出力される。パレット26Aの入出力データおよびパレット26Aに格納されるパレットデータは、上述した図18および図19と同様であるので、詳細な説明を省略する。

10

#### 【0441】

パレット26Aから出力されたRGBデータは、RGB/YCbCr変換回路26Bに供給されてYCbCrデータに変換され、動画データのデータ形式であるYCbCrデータに統一される。RGB/YCbCr変換回路26Bから出力されたYCbCrデータが乗算器27に入力される。

#### 【0442】

グラフィクスプレーン12'で用いられる画像データがPNG形式である場合には、画像データ中に、画素毎に不透明度データ $\alpha 2$ ( $0 \leq \alpha 2 \leq 1$ )を設定することができる。不透明度データ $\alpha 2$ は、乗算器27に供給される。乗算器27では、RGB/YCbCr変換回路26から入力されたYCbCrデータに対し、輝度信号Y、色差信号Cb、Crのそれぞれについて、不透明度データ $\alpha 2$ との乗算が行われる。乗算器27による乗算結果が加算器28の一方の入力端に入力される。また、不透明度データ $\alpha 2$ の補数( $1 - \alpha 2$ )が乗算器25に供給される。

20

#### 【0443】

2-16. デコーダモデルの別の例について

図69は、図34を用いて説明したプレーヤデコーダ100に対して、上述したグラフィクスオブジェクト200に連動可能な音声データ処理と、グラフィクスプレーン12'のカラーシステムを256色のパレットを用いた8ビットカラーマップアドレスとを適用した場合のプレーヤデコーダ100'の一例の構成を示す機能ブロック図である。なお、図69において、上述の図34と共通する部分には同一の符号を付し、詳細な説明を省略する。

30

#### 【0444】

上述の、クリップAVストリームに多重化されない音声データは、例えば入力チャンネル(1)のデータとして入力端101に入力され、スイッチ回路102および103を介してコンテンツバッファ105に供給される。一方、音声データが格納されたグラフィクスオブジェクト200が多重化されたクリップAVストリームは、入力端202に入力される。そして、PIDフィルタ110でグラフィクスオブジェクト200が振り分けられバッファTBn111Aに一旦溜め込まれ、スイッチ回路103を介してコンテンツバッファ105に供給される。

#### 【0445】

音声データの格納されていないグラフィクスオブジェクト200がクリップAVストリームに多重化されて入力端202から入力される。PIDフィルタ110でグラフィクスオブジェクト200を構成するトランスポートパケットが振り分けられ、バッファTBn111BまたはバッファTBn111Cに一旦溜め込まれる。バッファTBn111Bに溜め込まれたトランスポートパケットは、バッファBn112Bに供給され、PIDヘッダに基づきグラフィクスオブジェクト200がまとめられる。グラフィクスオブジェクト200は、スイッチ回路107を介してグラフィクスデコーダA500に供給される。バッファTBn111Cに溜め込まれたトランスポートパケットも、同様にしてバッファBn112Cを用いてグラフィクスオブジェクト200がまとめられ、スイッチ回路108を介してグラフィクスデコーダB501に供給される。

40

50



## 【0446】

グラフィクスデコーダA500およびB501では、供給されたトランスポートパケットのヘッダ情報を除去すると共に、当該トランスポートパケットに格納された画像データをデコードして字幕またはグラフィクスを表示するための画像データ、例えばビットマップデータとする。

## 【0447】

一方、コンテンツバッファ105に供給された、音声データが格納されたグラフィクスオブジェクト200に格納される画像データは、コンテンツバッファ105から取り出され、スイッチ回路107および108を介してグラフィクスデコーダA500およびB501にもそれぞれ供給される。

10

## 【0448】

この図69の例では、グラフィクスデコーダA500は、PNG形式の画像データをデコードし、グラフィクスデコーダB500は、JPEG形式の画像データをデコードする。なお、グラフィクスデコーダA500およびB501は、これに限らず、さらに他のデータ形式の画像データをデコードするものであってもよいし、複数の形式の画像データに対応するようにもできる。

## 【0449】

グラフィクスデコーダA500の出力は、スイッチ回路130の入力端130Bおよびスイッチ回路131の入力端131Bに供給され、スイッチ回路130および131を介してそれぞれサブピクチャプレーン502およびグラフィクスプレーン503に供給される。

20

## 【0450】

マルチメディアエンジン106は、上述の図34の構成に加えてサウンドプレーヤ106Dを有する。バッファ109は、上述の図34の構成に加えてサウンドバッファ109Eを有する。サウンドプレーヤ106Dは、サウンドバッファ109Eを用いて、コンテンツバッファ105から読み出した音声データを復号化して、例えばリニアPCMオーディオデータにして出力する。サウンドプレーヤ106Dから出力された音声データは、プレゼンテーションプロセッサ141に供給され、オーディオデコーダ118から出力された音声データと混合されて、出力端142に導出される。

## 【0451】

例えばボタン画像をクリックした際に発生されるクリック音など、効果音となる音声データは、サウンドプレーヤ106Dによって再生される。音声データは、サウンドバッファ109Eに蓄積され、サウンドプレーヤ106Dにより再生される。サウンドプレーヤ106Dは、例えば音声データのソフトウェア的な再生処理を行う。

30

## 【0452】

マルチメディアエンジン106では、例えばコードバッファ104に蓄えられたECMAスクリプトを読み出して解析し、コードバッファ104からの他のECMAスクリプトやHTML文書の読み出し、コンテンツバッファ105からの画像データや音声データの読み出しなどを行う。コンテンツバッファ105に格納された音声データは、コンテンツバッファ105に格納された他のデータと同様、上述したように、コンテンツバッファ105に保持しておくことができる。

40

## 【0453】

さらに、マルチメディアエンジン106により、ユーザからの、リモートコントロールコマンドやポインティングデバイスなどによる入力を受け取られ、所定に処理される。マルチメディアエンジン106では、このユーザ入力に基づく処理結果や各スクリプトに応じてコントロール信号を生成する。このコントロール信号は、さらに、グラフィクスデコーダA500およびB501、オーディオデコーダ118、MPEGビデオデコーダ120およびシステムデコーダ121にも供給される。

## 【0454】

グラフィクスレンダラ106Cで処理された画像データは、スイッチ回路130および

50

131をそれぞれ介してサブピクチャプレーン502およびグラフィクスプレーン503に供給される。サブピクチャプレーン502およびグラフィクスプレーン503は、例えばフレームメモリからなり、図66を用いて説明した字幕プレーン11およびグラフィクスプレーン12'にそれぞれ対応する。

#### 【0455】

なお、ここでは、グラフィクスレンダラ106Cからサブピクチャプレーン502およびグラフィクスプレーン503に供給される画像データは、例えばPNG形式またはJPEG形式の画像データをグラフィクスレンダラ106Cでデコードした後のビットマップデータであるものとする。

#### 【0456】

マルチメディアエンジン106は、さらに、プレゼンテーションプロセッサ139に対して、サブピクチャプレーン502、グラフィクスプレーン503および動画像プレーン135を切り換える制御信号を供給し、プレゼンテーションプロセッサ141に対して、オーディオストリーム出力を制御するような制御信号を供給する。

#### 【0457】

サブピクチャプレーン502上の画像データは、上述の図68におけるパレットテーブル22に対応するパレット504に供給され、256色からなるパレットに対してインデックスによる参照がなされ、RGBデータが出力されると共に、不透明データa1が出力される。RGBデータは、上述の図68におけるRGB/YCbCr変換回路29に対応するRGB/YCbCr変換回路506に供給され、カラーシステムがRGB(4:4:4)からYCbCr(4:4:4)に変換される。RGB/YCbCr変換回路506から出力されたYCbCrデータは、プレゼンテーションプロセッサ139に供給される。

#### 【0458】

グラフィクスプレーン503上の画像データは、上述の図68におけるパレットテーブル26Aに対応するパレット505に供給され、256色からなるパレットに対してインデックスによる参照がなされ、RGBデータが出力されると共に、不透明データa2が出力される。RGBデータは、上述の図68におけるRGB/YCbCr変換回路26Bに対応するRGB/YCbCr変換回路507に供給され、カラーシステムがRGB(4:4:4)からYCbCr(4:4:4)に変換される。RGB/YCbCr変換回路507から出力されたYCbCrデータは、プレゼンテーションプロセッサ139に供給される。

#### 【0459】

プレゼンテーションプロセッサ139には、さらに、動画像プレーン135上の動画像データがアップ/ダウンコンバータ138を介して供給される。

#### 【0460】

プレゼンテーションプロセッサ139は、上述したように、字幕プレーン11(サブピクチャプレーン502の画像データによる不透明度 $\alpha_1$ と、グラフィクスプレーン12'(グラフィクスプレーン503)による不透明度 $\alpha_2$ とを用いたアルファブレンディング処理を行う。この処理により、動画プレーン10、字幕プレーン11およびグラフィクスプレーン12'の画像データが合成される。また、プレゼンテーションプロセッサ139は、画像データに対してリアルタイムでエフェクト処理を行うこともできる。このように、プレーン間の合成処理やエフェクト処理がなされた画像データが出力端140に導出される。

#### 【0461】

2-17. デコーダモデルのさらに別の例について

図70は、上述した音声データの第3の格納方法を適用した場合のプレーヤデコーダ100の一例の構成を示す。なお、図70において、上述した図34または図69と共通する部分には同一の符号を付し、詳細な説明を省略する。上述したように、音声データの第3の格納方法では、ボタン状態に対応する効果音を再生する複数の音声データを、まとめて1つのファイル「HdmvSound.bdmv」とし、ディレクトリBD-VAの配下に格納してい

10

20

30

40

50

る。このファイル「HdmvSound.bdmv」は、ディスクのイニシャルアクセス時に予め読み込まれ、プレーヤ内部のメモリに格納される。

【0462】

プレーヤ内部のメモリは、バッファ109内のサウンドバッファ109Eを用いることができる。例えば、ディスクのイニシャルアクセス時に読み出された音声データファイルが端子101から入力され、スイッチ回路102およびコンテンツバッファ105を介してサウンドバッファ109Eに供給される。コードバッファ104に溜め込まれたプログラムコードの実行に伴い、サウンドバッファ109Eから必要な音声データが読み出され、端子710から入力される。

【0463】

また、端子202にMPEG ESで入力されたリアルタイムストリームは、PIDフィルタ110に供給される。PIDに基づき、トランスポートパケットがインタラクティブグラフィクスストリーム700が格納されているパケットであるとされれば、当該トランスポートストリームは、バッファ111Aに一旦溜め込まれ、所定のタイミングで読み出されて入力端103Bが選択されたスイッチ回路103に入力され、スイッチ回路103を介してコンテンツバッファ105に格納される。

【0464】

コンテンツバッファ105から、インタラクティブグラフィクスストリーム700内のインタラクティブコンポジションセグメント701が読み出され、マルチメディアエンジン106に供給される。また、コンテンツバッファ105から、ボタン画像データ702、702、・・・が読み出され、スイッチ回路108を介してグラフィクスデコーダB501に供給される。

【0465】

端子710から入力された音声データは、バッファ711に一旦溜め込まれ、ミキサ712においてオーディオデコーダ118から出力された音声データと所定の比率で混合されて、プレゼンテーションプロセッサ141に供給される。ミキサ712における混合比は、例えばユーザ入力に応じたサウンドプレーヤ106Dの制御に基づき、プレーヤデコーダ100側で設定可能である。

【0466】

なお、図70では、便宜上、バッファ711およびミキサ712を独立した構成のように表現しているが、バッファ711およびミキサ712は、実際には、プレゼンテーションプロセッサ141に内蔵されるものである。したがって、上述した図62との対応を考えると、バッファ711がバッファ450Bに対応し、ミキサ712がオーディオミキサ231に対応する。また、図62におけるバッファ450Aは、プレゼンテーションプロセッサ141に内蔵される。

【0467】

また、音声データが圧縮符号化されてディスクに記録されている場合、ディスクから読み出された音声データは、圧縮符号が復号化され、リニアPCMオーディオデータとされてから、バッファ711に溜め込まれる。例えば、サウンドプレーヤ106Dは、ディスクから読み出された、圧縮符号化された音声データをデコードしてサウンドバッファ109Eに格納する。音声データのデコードには、オーディオデコーダ118を用いることができる。ソフトウェア的にデコード処理を行ってもよい。

【0468】

上述では、図70のプレーヤデコーダモデルにおいて、効果音を再生する音声データは、ディスクに格納された映画などの主コンテンツの再生に先立って、ディスクからプリロードされて、プレーヤデコーダの内部メモリに格納されると説明した。ここで、上述した図62のようなディレクトリおよびファイル構造の場合、効果音を再生する音声データが格納されているファイル「HdmvSound.bdmv」は、ディレクトリBDMVに対して1個しか存在できない。プレーヤは、ディスクへのイニシャルアクセス時に、効果音の音声データファイルをプレーヤの内部メモリにプリロードする。

10

20

30

40

50

## 【0469】

これはこの例に限らず、別の例として、図71に一例が示されるように、プレイリストに対して効果音再生のためのデータを付随させる方法が考えられる。この場合は、既に説明した、図10に一例が示されるサブプレイアイテムを利用して、ボタンに対する効果音を再生する音声データを参照する方法を用いることができる。例えば、サブプレイアイテムに対して、当該サブプレイアイテムで参照される音声データが格納されるファイル名（ファイル「HdmvSoundxx.bdmv」）を記述する。

## 【0470】

あるいは、効果音のための音声データが格納されているファイルへの参照情報を、プレイリストの属性情報として付加する方法として、図9中のブロックUIAppInfoPlayList() 10を利用してよい。この場合は、ブロックUIAppInfoPlayList()のシンタクスに対して、効果音のための音声データが格納されているファイル名「HdmvSoundxx.bdmv」を記述するフィールドを設ければよい。

## 【0471】

プレーヤデコーダは、プレイリストの再生に先立って、効果音のための音声データが格納されているファイルへの参照情報（例えばサブプレイアイテム）により参照される音声データを、プレーヤの内部メモリにプリロードする。

## 【0472】

図72は、この別の例における一例のファイルの管理構造を示す。この図72の例では、ディレクトリSTREAMの下に、効果音を再生する音声データが格納されるファイル「HdmvSound01.bdmv」、ファイル「HdmvSound02.bdmv」、・・・が置かれる。ファイル「HdmvSound01.bdmv」、ファイル「HdmvSound02.bdmv」、・・・のそれぞれは、図60および図61を用いて説明したファイル「HdmvSound.bdmv」と同一の構成を有し、例えばデータ部602に格納される音声データがそれぞれ異なる。このように、効果音を再生する音声データが格納されるファイルを、ディスク上に複数存在させることも可能である。上述のように、効果音のための音声データが格納されるファイルへの参照情報（例えばサブプレイアイテム、あるいは、プレイリストの属性情報であるブロックUIAppInfoPlayList()）によって、音声データが格納されるファイルが参照される。この別の例の場合、効果音を再生する音声データの種類を、プレイリスト単位で変更することが可能である。

## 【0473】

## 3. 発明の実施の第2の形態

次に、この発明の実施の第2の形態について説明する。上述の実施の第1の形態では、ディスクに対する再生制御を指示する再生プログラムをECMAスクリプトおよびHTMLを用いて記述した。これに対して、この実施の第2の形態では、再生プログラムを、ECMAスクリプトおよびHTMLとは異なる、独自のデータ構造、記述言語を用いて表現する。この発明の実施の第2の形態で用いられるシナリオ記述言語は、DVDビデオのナビゲーションコマンドを基に変更を加えたものに、字幕画像、ボタン画像用の独自の表示制御コマンドをさらに加えた命令群を定義する。これにより、Blu-ray Discに用いて好適な、より柔軟なメニュー画面表示を行うことを可能としている。

## 【0474】

## 3-1. シナリオについて

この実施の第2の形態では、独自に定義したシナリオ記述言語を用いるため、例えば上述の実施の第1の形態のECMAスクリプトを用いた場合に比べ、イベントハンドラの定義が不要となる。例えば、この独自に定義したシナリオ記述言語は、シナリオを実行するに当たって必要なイベントを予め言語として定義することができるため、汎用なECMAスクリプトを用いる場合のように、シナリオの実行に特に適合したイベントをプログラム内で定義する必要が無い。

## 【0475】

この独自に定義したシナリオ記述言語を用いることで、例えば上述した図21で一例を示したシナリオの内部構造は、グローバルなイベントハンドラ定義71とローカルなイベ 50

ントハンドラ定義72とが不要とされ図73に示されるシナリオ70'のようになる。シナリオ70'の先頭の、エントリプレイリストと記述されるプレイリスト73Aは、例えばディスクがプレーヤにローディングされてシナリオが再生された際に最初に実行されるプレイリストである。このメニュー画面に対して所定の指示がなされると、続けてプレイリスト73Bが再生され、再生が終了されるとプレイリスト73Cが再生される。この図73の例では、プレイリスト73Cの再生によりグラフィクスデータ74Aが読み出されて、ユーザにストーリーの分岐選択を促す画面80Aが表示される。

【0476】

以降、この発明の実施の第1の形態での説明において図21を用いて上述したようにして、シナリオ70'が再生される。プレイリスト中にマークが設定されている場合には、  
10 このマークの位置に応じたプレイリストの分岐や合流などが可能とされる。

【0477】

シナリオの再生において、マークの検出、ユーザからの入力、プレーヤの動作変更の検出がされた時の動作は、プレイリスト毎にコマンド列（プログラム）が用意されており、そのプログラムをプレーヤが実行することにより行われる。

【0478】

例えば、上述のシナリオ70'内のプレイリスト73A～73Mの何れが再生中であっても、リモートコントロールコマンドに設けられたメニューボタンが押されたら、シナリオの一覧が表示されるメニュー画面60（図20参照）が表示される、すなわち、メニュー画面60を表示するためのプレイリストに再生処理が移行される動作を実現したい場合  
20 について考える。この場合、リモートコントロールコマンドのメニューボタンが押されたときに発生するイベント（メニューボタン押下イベント）に対応するイベントハンドラとして、メニュー画面60を表示させるプレイリストに処理が移行される命令を、シナリオデータの一つとして記述しておく。

【0479】

### 3-2. シナリオの分類

シナリオはBD-AVディレクトリに一つ定義され、一つまたは二つ以上のプレイリストから構成される。ここで、シナリオの構成の分類について図74および図75を用いて説明する。シナリオの構成は、プレイリスト同士の接続を基に考えると、図74に示されるように、（1）シングルプレイリスト、（2）シーケンシャルプレイリスト、（3）マルチプレイリスト、の3種類に大別することができる。  
30

【0480】

（1）のシングルプレイリストは、図75Aに一例が示されるように、単一のプレイリストで構成されるシナリオである。タイムラインが定義可能で、シナリオ再生中の割り込みが無い。コンテンツが映画であれば、例えばディスクのローディング後に映画本編だけが再生されるようなシナリオである。

【0481】

（2）のシーケンシャルプレイリストは、図75Bに一例が示されるように、複数のプレイリストが分岐無しに直列的に並べられて構成されるシナリオである。各プレイリストは、プレイリストの末尾と次のプレイリストの先頭とが接続されるように並べられる。各  
40 プレイリストを通じてのタイムラインが定義可能である。コンテンツが映画であれば、例えばメニュー画面と映画本編との組み合わせからなるシナリオである。ディスクのローディング後にメニュー画面を表示させるプレイリストが実行され、メニュー画面に対して映画本編の再生が指示されると、次のプレイリストが実行され、映画の本編が再生される。

【0482】

（3）のマルチプレイリストは、図75Cに一例が示されるように、プレイリストの分岐や収束による接続を含むシナリオである。プレイリストを通じてのタイムラインを定義することができず、例えば各プレイリスト内にタイムラインが定義される。マルチプレイリストによれば、ユーザ入力によって再生内容が変化するようなインタラクティブ性やゲーム性を実現することができる。コンテンツが映画の場合には、例えば同一シーンを様々  
50

なアングルで収録し、再生時にユーザの指示により選択して鑑賞できるようにしたマルチアングルを実現するシナリオである。

#### 【0483】

シナリオは、後述するように、BD A Vディレクトリに一つ定義される。ユーザに対しては、シナリオをより細分化した単位を見せる必要がある。但し、プレイリストの単位がユーザが認識する単位と一致するとは限らない。例えば、一つのプレイリストに3本の映画が収録されているときには、各映画の頭出し点をユーザに見せる必要がある。このように、プレイリストの構造とは独立した頭出し点（エントリポイント）を、以下では、タイトルおよび／またはチャプタと呼ぶことにする。

#### 【0484】

図76を用いてタイトルおよびチャプタについて説明する。タイトルは、シナリオ中の任意の再生開始点を表す。この図76の例では、プレイリスト470Aの先頭にタイトル1に設定されている。タイトル2は、プレイリスト470Dの途中に設定されている。プレイリスト470Aの先頭からタイトル2の直前までがタイトル1とされる。チャプタは、タイトルをさらに細分化した単位で、これもユーザが再生開始点として認識できる単位となる。タイトル1がチャプタによりさらに細分化される。図76の例では、タイトル1に対してチャプタ1、2および3が設定され、タイトル1が3つに細分化されている。また、図76に示されるように、タイトルおよびチャプタの位置は、プレイリストの途中を指定することもできる。

#### 【0485】

### 3-3. 仮想プレーヤモデルについて

この発明の実施の第2の形態によるシナリオ記述言語は、この発明の実施の第1の形態において図22を用いて既に説明した、BD仮想プレーヤモデルにより同様に再生することができる。BD仮想プレーヤ30は、ディスクのローディング後、ディスクに記録された、この実施の第2の形態による独自に定義したシナリオ記述言語により記述されたシナリオを、PBCプログラム40として読み込み、シナリオの記述に従い動作する。このとき、BD仮想プレーヤ30の動作中に発生されたイベントに応じて、BD仮想プレーヤ30の動作が制御される。

#### 【0486】

シナリオは、プレーヤの動作を指示するコマンドが並んだプログラムを含むコマンドを配置する領域として、グローバルコマンド領域とローカルコマンド領域とを有する。

#### 【0487】

グローバルコマンド領域は、シナリオ全体に有効なプログラムであるグローバルコマンドが格納されている。グローバルコマンド領域には、例えば、ディスクがプレーヤに挿入されたときに、プレーヤによって最初に自動的にプレーヤに実行されて、パラメータの初期化を行い、メニュー画面を構成するプレイリストにジャンプするようなプログラムが記述される。ローカルコマンド領域は、各プレイリストに関係するプログラムが格納される。この発明の実施の第2の形態では、ローカルコマンドは、さらにプレコマンド、プレイアイテムコマンド、ポストコマンドおよびボタンコマンドの4種類に分類される。

#### 【0488】

図77は、この発明の実施の第2の形態におけるシナリオで記述されるコマンドによるBD仮想プレーヤ30の動作を概略的に示す。図77Aは、ディスクのローディング時の動作の例を示す。上述したように、シナリオは、BD A Vディレクトリに対して一つが作られる。ディスクがプレーヤに挿入されディスクに対するイニシャルアクセスがなされると（ステップS30）、レジスタすなわち共通パラメータ32が初期化される（ステップS31）。そして、次のステップS32で、プログラムがディスクから読み込まれて実行される。なお、イニシャルアクセスは、ディスク挿入時のように、ディスクの再生が初めて行われることをいう。

#### 【0489】

この、ディスクローディング時に最初に読み込まれ、実行されるコマンド群（プログラ

10

20

30

40

50

ム)を、グローバルコマンドと称する。グローバルコマンドは、例えば宣伝用映像(トレーラー)やメニュー画面を構成するプレイリストへのジャンプ命令が記述されており、その命令通りにプレーヤがプレイリストを再生していくことになる。

#### 【0490】

図77Bは、プレーヤ30が停止状態からユーザにより例えばプレイキーが押下され再生が指示された場合の動作の例を示す。これは、上述の図23における状態Bから状態Aへの状態遷移に相当する。最初の停止状態(ステップS40)に対して、ユーザにより、例えばリモコンなどを用いて再生が指示される(UOP: User Operation)。再生が指示されると、まず、レジスタすなわち共通パラメータ32が初期化され(ステップS41)、次のステップS42で、プレイリストの再生フェイズに移行する。

10

#### 【0491】

プレイリストの再生フェイズにおけるプレイリストの再生について、図78を用いて説明する。図78Aは、プレイリストが単一のプレイアイテムからなる場合の例である。プレイリストは、プレコマンド領域、プレイアイテムコマンド領域およびポストコマンド領域の3箇所にプログラムが配される。プレイリストの再生フェイズに移行すると、まずプレコマンド領域のプレコマンドが実行され(ステップS10)、プレコマンドの実行が終了されると、プレイリストを構成するプレイアイテムの再生フェイズに移行する(ステップS11)。プレイアイテムの再生フェイズでは、プレイアイテムにおいて開始点および終了点で指定されたストリームが再生される(ステップS110)。終了点までの再生が終了した時点で、プレイアイテムコマンドが実行される(ステップS111)。プレイアイテムコマンドが実行されると、次にポストコマンド領域のポストコマンドが実行され(ステップS12)、プレイリストの再生が終了される。

20

#### 【0492】

なお、ポストコマンドは、一般的な使用においては、次に再生すべきプレイリストへのジャンプ命令や、メニュー画面を構成するプレイリストへのジャンプ命令が記述される。ジャンプ命令が無い場合には、そこで再生が停止し、プレーヤは、停止状態(図23の状態B)に遷移される。

#### 【0493】

図78Bは、プレイリストが複数のプレイアイテムを含む場合の例である。この場合でも、プレイリスト中にプレコマンド領域、プレイアイテムコマンド領域およびポストコマンド領域が配され、それぞれプログラムが格納される。複数のプレイアイテムを含む場合、プレイアイテムコマンド領域では、プレイアイテム数分のプレイアイテムストリームおよびプレイアイテムコマンドが、時系列に従って配置される。

30

#### 【0494】

プレイリストが複数のプレイアイテムを含む場合でも、プレイリストの再生フェーズに移行すると、まず、プレコマンドが実行される(ステップS10)。次のプレイアイテムの再生フェイズでは、プレイアイテムの開始点から終了点までのストリームの再生と、プレイアイテムコマンドの実行とが、プレイリストが含むプレイアイテムの数だけ実行される。図78Bの例では、最初のプレイアイテムストリームが再生され(ステップS110-1)、対応するプレイアイテムコマンドが実行される(ステップS111-1)。次に、図示は省略するが、2番目のプレイアイテムストリームが再生され(ステップS110-2)、対応するプレイアイテムコマンドが実行される(ステップS111-2)。これをプレイアイテムの数だけ行い、最後のプレイアイテムストリームが再生され(ステップS110-n)、対応するプレイアイテムコマンドが実行されると(ステップS111-n)、プレイアイテムの再生フェイズが終了する。プレイアイテムの再生フェイズが終了すると、ポストコマンドが実行され(ステップS12)、このプレイリストの再生フェーズが終了される。

40

#### 【0495】

この発明の実施の第2の形態では、BD仮想プレーヤ30ならびに、BD仮想プレーヤ30上で実行されるシナリオ、プレイリストおよびプレイアイテムを、階層的に考える

50



ことができる。すなわち、図79Aに一例が示されるように、BD仮想プレーヤ層600の上に1つのシナリオ層601があり、シナリオ層601の上に、1または複数のプレイリストを含むプレイリスト層602がある。プレイリスト層602の上には、プレイアイテム(PI)層603があり、プレイリスト毎に1または複数のプレイアイテムを持つことができる。

【0496】

このような階層的な構造とすることで、プレイリストおよびプレイアイテムは、シナリオ層601を介してBD仮想プレーヤにより実行されることになる。したがって、シナリオにプレイリストに対する制御コマンドを記述することで、プレイリストの分岐などの制御が容易に実現される。これは、プレイアイテムに関しても同様である。この様子を図7 10  
9Bに示す。

【0497】

#### 3-4. シンタクスについて

次に、この発明の実施の第2の形態によるシナリオを記述するコマンドやデータベースのディスクへの格納方法について説明する。この発明の実施の第2の形態においてBD A Vフォーマットに対して拡張した機能の実現に必要なデータは、シナリオファイル「scenario.pbc」に記述されるものとする。シナリオファイル「scenario.pbc」は、上述の図4に示した「Blu-ray Disc Rewritable Format Ver1.0 part3」で規定された、記録媒体に記録されるファイルの管理構造に対し、ディレクトリBD A Vの配下に置かれる。この様子を図80に示す。

20

【0498】

図81は、シナリオを記述するためのシナリオファイル(scenario.pbc)の一例の構造を表すシンタクスを示す。ファイル「scenario.pbc」は、先頭から、ファイル識別記号、バージョン番号、ブロックの開始アドレスが記述され、その後に、機能別の情報毎にブロックが構成される。フィールドtype\_indicatorは、32ビットのデータ長を有し、このファイルがシナリオを記述したファイルであることを表す特定の文字列が記述される。フィールドversion\_numberは、32ビットのデータ長を有し、バージョン番号が記述される。フィールドTitleEntry\_start\_addressは、データ長が32ビットの符号無しの整数で、ブロックTitleEntry()が開始される位置をファイル「scenario.pbc」の先頭からの相対バイト数で表した値が記述される。同様に、フィールドScenrio\_start\_addressは、データ長が 30  
23ビットの符号無しの整数で、ブロックScenario()が開始される位置を、ファイル「scenario.pbc」の先頭からの相対バイト数で表した値が記述される。

【0499】

ブロックGlobalCommand()は、ファイルの先頭から41バイト目の固定位置から始まる。ブロックGlobalCommand()には、上述した、イニシャルアクセス時(ディスク挿入時のようにディスクの再生が始めて行われる時)に実行されるプログラムが記述される。上述のグローバルコマンドは、このブロックGlobalCommand()に格納される。GlobalCommand()の後には任意の数のpadding\_wordが入り、ブロック間に空きを持たせることができるようになっている。

【0500】

ブロックTitleEntry()は、シナリオ内の頭出し点が羅列される。シナリオは、BD A Vディレクトリに対して一つ作られ、BD A Vディレクトリの配下にある複数のプレイリストを結び付けて再生順序を定義する。一方、ユーザから見た場合、シナリオは、必ずしも一つの映像・音声単位として見える訳ではなく、複数の「タイトル」から構成されているように見える。

【0501】

例えば、一枚のディスクに映画が3本収録されている場合、再生順序を定義するシナリオは、ディスクに一つのみ、存在するが、ユーザには、当該ディスクに3つのタイトルが収録されているように見える。あるいは、その3つのタイトルのリストを表示し、ユーザに選択させるタイトルメニュー画面も含めて、4つのタイトルに分割されているように見 50

える。なお、メニュー画面もユーザにとっては一まとまりの映像・音声単位であるので、この発明の実施の第2の形態では、タイトル的一种としてみなすことにする。

#### 【0502】

このように、プレイリストの結びつきを定義するシナリオという単位と、ユーザが一まとまりの映像・音声として認識する単位には違いがあるため、上述のように、シナリオ内に頭出し点を定義しておく必要がある。このシナリオ内の頭出し点を、タイトルエントリと称する。

#### 【0503】

ブロックScenario()は、これまで説明してきた「シナリオ」が記述される。プレイリストの再生順序の情報や、プレイリスト毎に持つローカルコマンド領域などがブロックScenario()に格納される。

#### 【0504】

図82は、上述のブロックGlobalCommand()の一例の構造を表すシンタクスを示す。フィールドlengthは、データ長が32ビットの符号無し整数で、このフィールドlengthの直後からブロックGlobalCommand()の終わりまでの長さがバイト表記で記述される。フィールドnumber\_of\_commandsは、その後に続くフィールドcommandの数を表す。フィールドcommandは、データ長が32ビットで表される値であり、後述するプレーヤのパラメータのセット、指定のプレイリストの再生開始、演算などの命令が記述される。

#### 【0505】

図83は、ブロックTitleEntry()の一例のデータ構造を表すシンタクスを示す。フィールドlengthは、データ長が32ビットの符号無し整数で、このフィールドlengthの直後からブロックTitleEntry()の終わりまでの長さがバイト表記で記述される。フィールドnumber\_of\_titlesは、その直後のforループ中に記述されるタイトル頭出し点の数を表す。フィールドEntry\_PlayList\_file\_nameは、タイトルの頭出し点が含まれるプレイリストのプレイリストファイル名が記述される。フィールドTitle\_name\_character\_setは、次のフィールドTitle\_nameを表現するキャラクターセットを表す。次のフィールドTitle\_nameは、頭出し点に付けられた名前の文字列が記述される。

#### 【0506】

フィールドnumber\_of\_Chaptersは、その直後のforループ内に記述されるチャプタの数が記述される。チャプタは、上述したように、タイトルを細分化するもので、チャプタ点もタイトルと同様にユーザから見えており、シナリオ内の頭出し点として用いられる。フィールドTitle\_numberは、データ長が16ビットの符号無し整数で、forループ中の現在のループカウンタに対応するチャプタ（以下、このチャプタ）が属するタイトルの番号を表す。フィールドchapter\_entry\_PlayList\_file\_nameは、このチャプタが指しているプレイリストのプレイリストファイルのファイル名を表す。フィールドchapter\_ref\_to\_PlayItem\_idは、このチャプタが指しているプレイアイテムの番号を表す。フィールドchapter\_time\_stampは、このチャプタが指しているプレイアイテム上の時刻を表す。

#### 【0507】

ブロックTitle\_Menu\_Call\_PL()は、ユーザがタイトル表示を指示したときに表示されるメニューを構成するプレイリストを指定する。ユーザによるタイトル指定は、例えばプレーヤを遠隔操作するリモコンに設けられた、タイトルメニューキーを操作することでなされる。タイトルメニューは、シナリオに一つ存在し、タイトル頭出し点をユーザに提示するためのメニューである。ユーザが例えばリモコンを操作してタイトルメニュー上で所望のタイトルを選択すると、プレーヤは、上述したフィールドnumber\_of\_titlesの直後のforループ内に記述された情報に基づくタイトルエントリのリストから再生すべきプレイリストを知り、そのプレイリストの再生を開始する。

#### 【0508】

ブロックTitle\_Menu\_Call\_PL()において、フィールドflagsは、タイトルメニューに関する属性情報が記述される。フィールドTitleMenu\_entry\_PlayList\_file\_nameは、タイトルメニューを構成するプレイリストあるいはプレイリスト群の入り口となるプレイリスト

を指定する。フィールドTitleMenu\_ref\_to\_PlayItem\_idは、プレイリスト中の特定のプレイアイテムからタイトルメニューが開始されている場合、そのプレイアイテムの番号を指定する。プレイリストの先頭からの再生であれば、フィールドTitleMenu\_ref\_to\_PlayItem\_idの値は、[0]となる。

#### 【0509】

フィールドTitleMenu\_chapter\_time\_stampは、プレイアイテム中の時刻を表す。プレイアイテムの先頭からの再生であれば、プレイアイテムの先頭の時刻となる。フィールドUOP\_mask\_table()は、ユーザ操作を制限するためのデータが格納される。ここで制限されたユーザ操作は、仮にユーザがその操作を行ったとしても、プレーヤは反応してはならないものとされる。例えば、メニュー画面で早送りが実行されないようにする場合に、このフィールドUOP\_mask\_table()で、早送りのユーザ操作を無効にするような設定をしておく。

#### 【0510】

図84は、ブロックscenario()の一例の構造を表すシンタクスを示す。ブロックscenario()は、シナリオの情報、すなわちプレイリスト間のリンクを定義する。フィールドlengthは、このフィールドの直後からブロックScenario()の終わりまでの長さがバイト表記で記述される。フィールドnumber\_of\_PlayListsは、シナリオを構成するプレイリストの数を表す。フィールドnumber\_of\_PlayLists直後のforループ内に、このforループのループカウンタに対応するプレイリスト（以下、このプレイリスト）の情報が、フィールドnumber\_of\_PlayListsに記述されるプレイリスト数だけ、記述される、

#### 【0511】

以降は、このプレイリスト毎に持つデータである。フィールドflagsは、プレイリストに関する属性情報が記述される。フィールドflagsに記述される内容は、この発明の実施の第2の形態に直接的な関連が無いので、詳細は省略する。ブロックPL\_UOP\_mask\_table()は、ユーザ操作を制限する情報が格納される。このブロックPL\_UOP\_mask\_table()では、ユーザ操作の制限をプレイリスト毎に設定することができる。後述するコマンドUOP\_mask\_table()およびこのフィールドPL\_UOP\_mask\_table()の両方で許可されているユーザ操作だけが、このプレイリスト再生中に実行できるユーザ操作となる。但し、最終的には、後述するブロックPL\_UOP\_mask\_table()内に記述されるデータも含めて、ユーザ操作が可能か否かが判定される。

#### 【0512】

フィールドParental\_levelは、このプレイリストに視聴者制限を加える場合に必要となる情報であり、このプレイリストが視聴可能な年齢や、視聴可能な年齢層を定めたレベルが記述される。

#### 【0513】

フィールドnumber\_of\_Pre\_Commandsは、このプレイリストを再生する前に実行されるプログラムを構成するコマンド（プレコマンド）の数を表す。プログラムは、後のフィールドPL\_Command(i)で表される領域に記述される。フィールドnumber\_of\_Post\_Commandsは、このプレイリストを再生した後に実行されるプログラム（ポストコマンド）を構成するコマンドの数を表す。プログラムは、後のフィールドPL\_Command(i)で表される領域に記述される。フィールドPre\_Commands\_start\_idは、プレイリストを再生する前に実行されるプログラムのコマンドテーブル中の開始番号を表す。番号は、フィールドPL\_Command(i)におけるパラメータ i に対応する。フィールドPost\_Commands\_start\_idは、プレイリストを再生した後に実行されるプログラムのコマンドテーブル中の開始番号を表す。番号は、フィールドPL\_Command(i)におけるパラメータ i に対応する。

#### 【0514】

フィールドnumber\_of\_PlayItemsは、このプレイリストを構成するプレイアイテムの数を表す。フィールドPI\_UOP\_mask\_table()は、ユーザ操作を制限する情報が格納される。フィールドPI\_UOP\_mask\_table()により、ユーザ操作の制限をプレイアイテムごとに設定できる。後述するコマンドUOP\_mask\_table()およびコマンドPL\_UOP\_mask\_table()と、このフィールドPI\_UOP\_mask\_tableの3種類のテーブルにおいて許可されているユーザ操作

だけが、このプレイアイテム再生中に実行できるユーザ操作となる。

#### 【0515】

フィールドPL\_Commands\_start\_idは、プレイアイテムの再生後に実行するコマンドの、コマンドテーブル中の開始番号を表す。番号は、フィールドPL\_Command(i)におけるパラメータiを表す。フィールドnumber\_of\_PL\_Commandsは、直後のforループにより表されるコマンドテーブルに格納されるコマンドの数を表す。フィールドPL\_Command(i)に格納されるコマンドには、順に番号iが割り当てられている。フィールドPL\_Command(i)は、一つのコマンドを表し、番号iは、上述したフィールドPre\_Commands\_start\_id、フィールドPost\_Commands\_start\_idおよびフィールドPL\_Commands\_start\_idなどから参照される。

#### 【0516】

### 3-5. コマンドについて

図85は、この発明の実施の第2の形態によるシナリオで用いられるコマンドの一例を示す。これらのコマンドは、上述のBD仮想プレーヤ30が有するメソッドである。なお、図85に例示されるコマンドは、この発明の実施の第2の形態によるシナリオで用いられるコマンドの一部であって、実際には、さらに多数のコマンドが定義され、上述の実施の第1の形態で説明したような、ユーザインターフェイスの多彩な機能を実現可能なようにされている。例えば、音声データの再生に関するコマンドや、ボタン画像表示に関するコマンドがさらに定義される。

#### 【0517】

再生開始位置指定に関するメソッドについて説明する。メソッドLinkPlayList(playListNumber)で、"playListNumber"で指定されたプレイリストの再生開始が指示される。メソッドLinkPlayItem(playListNumber,playItemNumber)で、指定のプレイリストの指定のプレイアイテムからの再生開始が指示される。"playItemNumber"は、"PlayItem\_id"であり、値が「0」から始まる。"playItemNumber"に値「0」を指定することで、当該プレイアイテムが属するプレイリストの先頭から再生される。

#### 【0518】

メソッドLink(position)(object)で、シナリオ内での移動が指示される。すなわち、現在移動中の箇所から前後のプレイリスト、プレイアイテムおよびチャプタに移動することがこのメソッドで指示される。なお、パラメータ"position"は、"prev"、"next"、"top"、"Parent"または"tail"の何れかの値を取り、パラメータ"object"で示される移動対象（プレイリスト、プレイアイテムまたはチャプタ）に対する移動方法が指示される。

#### 【0519】

メソッドExitで、シナリオ再生の停止が指示される。この場合、標準レジスタの値は、保持されない。メソッドRSMで、プレーヤ内のメモリに保存されているレジューム情報を呼び出し、レジスタにセットして再生を開始する。

#### 【0520】

プレーヤ状態の取得に関するメソッドについて説明する。メソッド getMenuDescription Language()で、メニュー表示の際に用いられる言語が取得される。メソッド getScenarioNumber()、メソッド getPlayListNumber()およびメソッド getChapterNumber()で、再生中のシナリオ番号、プレイリスト番号およびチャプタ番号がそれぞれ取得される。メソッド get tPlayerSupport()で、プレーヤのバージョン情報が取得される。

#### 【0521】

ビデオストリームに関するメソッドについて説明する。メソッド getVideoStreamAvailability()で、指定のビデオストリームが含まれているか否かが取得される。メソッド setVideoStreamNumber()で、デコードするビデオストリームが指定される。メソッド getVideoStreamNumber()で、選択中のビデオストリームの番号が取得される。メソッド getVideoStreamAttr()で、選択中のビデオストリームの属性が取得される。なお、ビデオストリームの属性は、例えば、そのビデオストリームの符号化方式、解像度、アスペクト比、アスペクト比が4:3のときのディスプレイモード、クローズドキャプションの有無である。メソッド setAngleNumber()で、アングル番号が指定される。メソッド getAngleNumber()で、

10

20

30

40

50

選択中のアングル番号が取得される。メソッドgetMaxVideoStream()で、ビデオストリームの最大数が取得される。

【0522】

このように定義されるシナリオファイル「scenario.pbc」に基づきディスクの再生を行うためには、例えば、上述の実施の第1の形態で図34、図69または図70を用いて説明した、プレーヤデコーダ100、100'または100"において、マルチメディアエンジン106に対して、これまでに説明した、実施の第2の形態によるシナリオ記述言語を解析するためのエンジンを追加すればよい。

【0523】

例えば図69に示されるプレーヤデコーダ100'において、図示されないドライブ装置にディスクが装填されると、まず、シナリオファイル「scenario.pbc」が再生される。再生されたシナリオファイル「scenario.pbc」は、プレーヤデコーダ100'に対してストアオブジェクトとして入力端101から入力され、スイッチ回路102を介してコードバッファ104に供給される。シナリオファイル「scenario.pbc」は、マルチメディアエンジン106内の上述のシナリオファイル解析エンジンにより解析され、解析結果に基づきシナリオデータが読み出され、動画プレーン10に表示するための動画データ、字幕プレーン11やグラフィクスプレーン12（またはグラフィクスプレーン12'）に表示するための画像データ、プレイリストファイル、シナリオファイルから呼び出される別のシナリオファイル、音声データなどがディスクから読み出される。

【0524】

4. その他

また、上述では、音声データの第1、第2および第3の格納方法がこの発明の実施の第1の形態に適用されるように説明したが、これはこの例に限定されず、音声データの第1、第2および第3の格納方法をこの発明の実施の第2の形態に適用させてもよい。

【0525】

さらに、上述では、この発明がBlu-ray Discに適用されるように説明したが、これは一例であって、この発明は、他の方式の大容量ディスク状記録媒体に対しても適用可能なものである。

【図面の簡単な説明】

【0526】

【図1】 AVストリームファイルの再生順序指定の仕組みを模式的に示す略線図である。

【図2】 クリップAVストリーム、クリップ情報、クリップ、プレイアイテムおよびプレイリストの関係を示すUML図である。

【図3】 複数のプレイリストから同一のクリップを参照する方法を説明するための略線図である。

【図4】 記録媒体に記録されるファイルの管理構造を説明するための略線図である。

【図5】 ファイル「info.bdav」の一例の構造を表すシンタクスを示す略線図である。

【図6】 ブロックUIAppInfoBDVAV()の一例の構造を表すシンタクスを示す略線図である。

【図7】 ブロックTableOfPlayLists()の一例の構造を表すシンタクスを示す略線図である。

【図8】 ファイル「#####.rp1s」および「#####.vp1s」の一例の構造を表すシンタクスを示す略線図である。

【図9】 ブロックUIAppInfoPlayList()の一例の構造を表すシンタクスを示す略線図である。

【図10】 ブロックPlayList()の一例の構造を表すシンタクスを示す略線図である。

【図11】 ブロックPlayItem()の一例の構造を表すシンタクスを示す略線図である。

【図12】 ブリッジクリップを説明するための略線図である。

【図13】 ブロックPlayListMark()の一例の構造を表すシンタクスを示す略線図である。

【図14】 ファイル「%%%%.clpi」の一例の構造を表すシンタクスを示す略線図である。

【図15】 この発明の実施の一形態で画像の表示系として用いられるプレーン構造の一例

を示す略線図である。

【図16】 動画プレーン、字幕プレーンおよびグラフィクスプレーンの一例の解像度および表示可能色を示す略線図である。

【図17】 動画プレーン、字幕プレーンおよびグラフィクスプレーンを合成する一例の構成を示すブロック図である。

【図18】 パレットの入出力データの一例を示す略線図である。

【図19】 パレットに格納される一例のパレットテーブルを示す略線図である。

【図20】 グラフィクスプレーンに表示されるメニュー画面の一例を示す略線図である。

【図21】 シナリオの一例の内部構造を示す略線図である。

【図22】 BD仮想プレーヤモデルについて説明するための略線図である。

10

【図23】 BD仮想プレーヤにおいて独自に定義される一例のイベントを示す略線図である。

【図24】 この発明の実施の一形態で定義される、BD仮想プレーヤが有するメソッドの一例を示す。

【図25】 この発明の実施の一形態で定義される、BD仮想プレーヤが有するメソッドの一例を示す。

【図26】 この発明の実施の一形態で定義される、BD仮想プレーヤが有するメソッドの一例を示す。

【図27】 ディスクに記録されるシナリオ構成の一例を示す略線図である。

【図28】 シナリオを構成する際に必要とされる一例のファイルを一覧して示す略線図である。

20

【図29】 シナリオがディスクに記録される際の一例のディレクトリ構造を示す略線図である。

【図30】 スクリプトファイルのより具体的な記述の例を示す略線図である。

【図31】 スクリプトファイルのより具体的な記述の例を示す略線図である。

【図32】 HTMLファイルのより具体的な記述の例を示す略線図である。

【図33】 スクリプトファイルのより具体的な記述の例を示す略線図である。

【図34】 プレーヤデコーダの一例の構成を示す機能ブロック図である。

【図35】 ボタン表示の一例の状態遷移図である。

【図36】 オブジェクトの種類の分類を説明するための図である。

30

【図37】 この発明の実施の一形態によるグラフィクスオブジェクトのデータ構造の例を示す。

【図38】 グラフィクスオブジェクトが分割されてPESパケットに格納される様子を示す略線図である。

【図39】 グラフィクスオブジェクトのデコードを行うグラフィクスオブジェクトデコーダモデルの一例の構成を示す機能ブロック図である。

【図40】 グラフィクスオブジェクト入力バッファ、PNGデコーダ、オブジェクトバッファおよびプレーンバッファにおける蓄積データ量の遷移の例を概略的に示す略線図である。

【図41】 プレーンの更新速度について説明するための図である。

40

【図42】 グラフィクスオブジェクトの一例の構造を表すシンタクスを示す略線図である。

【図43】 ブロックGlobalPaletteTable()の一例の構造を表すシンタクスを示す略線図である。

【図44】 命令グループDispCmds(i)に置かれる表示制御命令の例を一覧して示す略線図である。

【図45】 命令set\_display\_box(x1,y1,x2,y2)および命令set\_clipping\_box(a1,b1,a2,b2)を説明するための略線図である。

【図46】 座標軸の定義を説明するための略線図である。

【図47】 命令グループDispCmds(i)の記述とグラフィクスオブジェクトの表示変化の一

50

例を示す。

【図48】字幕表示が徐々に現れるフェードインの例を示す略線図である。

【図49】字幕のPNG画像がプレーン内を移動する例を示す略線図である。

【図50】字幕表示をスクロールさせる例を示す略線図である。

【図51】PNG画像の一部を表示する枠を設定し、PNG画像上でこの枠を移動させつつ、プレーン上での表示位置も移動させる例を示す略線図である。

【図52】命令play\_sound(sound\_id)が追加された命令グループDispCmds(i)の記述とグラフィクスオブジェクトの表示変化の一例を示す略線図である。

【図53】ボタン画像に音声データを割り当てたグラフィクスオブジェクトの一例のデータ構造を示す略線図である。

10

【図54】グラフィクスオブジェクトに音声データを格納する一例の方法を示す略線図である。

【図55】音声データの処理系を追加したグラフィクスオブジェクトデコーダモデルの一例の構成を示す機能ブロック図である。

【図56】音声データを加えた場合のグラフィクスオブジェクトの一例の構成を示すシンタクスを示す略線図である。

【図57】グラフィクスオブジェクトに音声データを格納しない場合の一例のデータの管理構造を示す略線図である。

【図58】グラフィクスオブジェクトに音声データを格納しない場合の音声データの再生についてより具体的に説明するための略線図である。

20

【図59】記録媒体に記録されるファイルの管理構造を説明するための略線図である。

【図60】ファイル「HdmvSound.bdmv」の一例のシンタクスを示す略線図である。

【図61】ファイル「HdmvSound.bdmv」の構成を示す略線図である。

【図62】第3の格納方法による効果音再生の一例のアプリケーションイメージを示す略線図である。

【図63】インタラクティブグラフィクスストリームの一の構成を概略的に示す略線図である。

【図64】インタラクティブコンポジションセグメントの一の構成を示す略線図である。

【図65】インタラクティブコンポジションセグメントにおけるフィールドneighbour\_infoを説明するための図である。

30

【図66】この発明の実施の一形態で画像の表示系として用いられるプレーン構造の他の例を示す略線図である。

【図67】動画プレーン、字幕プレーンおよびグラフィクスプレーンの他の例の解像度および表示可能色を示す略線図である。

【図68】他の例により動画プレーン、字幕プレーンおよびグラフィクスプレーンを合成する一例の構成を示すブロック図である。

【図69】プレーヤデコーダの一例の構成を示す機能ブロック図である。

【図70】プレーヤデコーダの一例の構成を示す機能ブロック図である。

【図71】サブプレイアイテムを用いて音声データファイルを参照することを説明するための図である。

40

【図72】記録媒体に記録されるファイルの管理構造を説明するための略線図である。

【図73】この発明の実施の第2の形態による独自に定義されたシナリオ記述言語を用いた場合のシナリオの一例の内部構造を示す略線図である。

【図74】シナリオの構成の分類を説明するための略線図である。

【図75】シナリオの構成の分類を説明するための略線図である。

【図76】タイトルおよびチャプタについて説明するための略線図である。

【図77】この発明の実施の第2の形態におけるシナリオで記述されるコマンドによるBD仮想プレーヤ30の動作を概略的に示すフローチャートである。

【図78】プレイリストの再生動作を説明するためのフローチャートである。

50



【図 7 9】シナリオの一例の階層構造を示す略線図である。

【図 8 0】記録媒体に記録されるファイルの管理構造を説明するための略線図である。

【図 8 1】シナリオを記述するためのシナリオファイル(scenario.pbc)の一例の構造を表すシンタクスを示す略線図である。

【図 8 2】ブロックGlobalCommand()の一例の構造を表すシンタクスを示す略線図である。

【図 8 3】ブロックTitleEntry()の一例のデータ構造を表すシンタクスを示す略線図である。

【図 8 4】ブロックscenario()の一例の構造を表すシンタクスを示す略線図である。

【図 8 5】この発明の実施の第 2 の形態によるシナリオで用いられるコマンドの一例を示す略線図である。

【図 8 6】従来技術による一例の DVD メニュー画面を示す略線図である。

【図 8 7】リモコン操作により選択状態のボタンを変更する様子を示す略線図である。

【図 8 8】決定ボタンの操作によりボタンが実行状態を表す色に変更される様子を示す略線図である。

【図 8 9】従来技術によるメニュー画面の構成の例を示す略線図である。

【図 9 0】従来技術によるサブピクチャの一例のデータ構成を示す略線図である。

【符号の説明】

【 0 5 2 7 】

- 1 0 動画プレーン
- 1 1 字幕プレーン
- 1 2 グラフィクスプレーン
- 2 2 バレット
- 3 0 BD 仮想プレーヤ
- 3 1 プレーヤコマンド
- 3 2 共通パラメータ
- 4 0 プレイバックコントロールプログラム
- 4 1 メソッド
- 6 0 メニュー画面
- 7 0 シナリオ
- 7 3 A ~ 7 3 M プレイリスト
- 1 0 0, 1 0 0', 1 0 0" プレーヤデコーダ
- 1 0 4 コードバッファ
- 1 0 5 コンテンツバッファ
- 1 0 6 マルチメディアエンジン
- 1 0 9 バッファ
- 1 1 0 P I D フィルタ
- 1 1 6 サブピクチャデコーダ
- 1 1 7 静止画像デコーダ
- 1 1 8 オーディオデコーダ
- 1 2 0 M P E G ビデオデコーダ
- 1 3 2 サブピクチャプレーン
- 1 3 3 グラフィクスプレーン
- 1 3 4 静止画像プレーン
- 1 3 5 動画画像プレーン
- 2 2 6 P N G デコーダバッファ
- 2 2 7 P N G デコーダ
- 2 2 8 オブジェクトバッファ
- 2 2 9 プレーンバッファ
- 2 3 1 オーディオミキサ

20

30

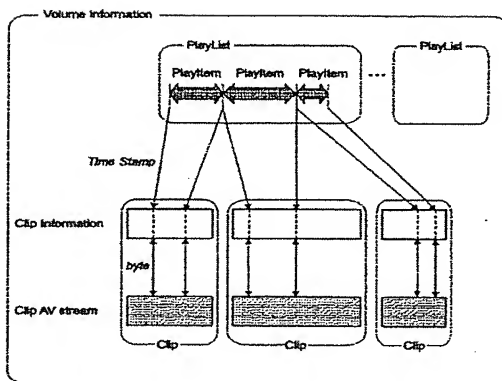
40

50

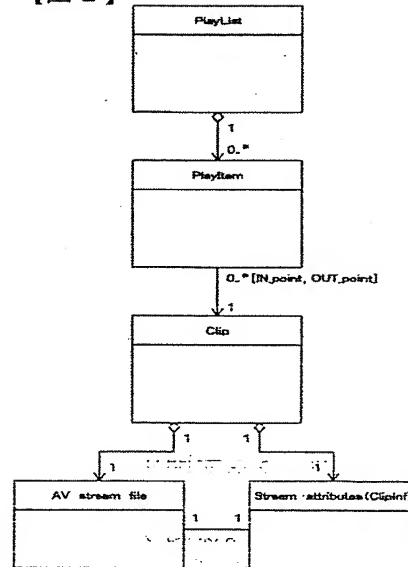
5 0 0 グラフィクスデコーダ A  
 5 0 1 グラフィクスデコーダ B  
 5 0 2 サブピクチャプレーン  
 5 0 3 グラフィクスプレーン  
 6 0 1 ループ部  
 6 0 2 音声データ部  
 7 0 0 インタラクティブグラフィクスストリーム  
 7 0 1 インタラクティブコンポジションセグメント  
 7 1 1 バッファ  
 7 1 2 ミキサ

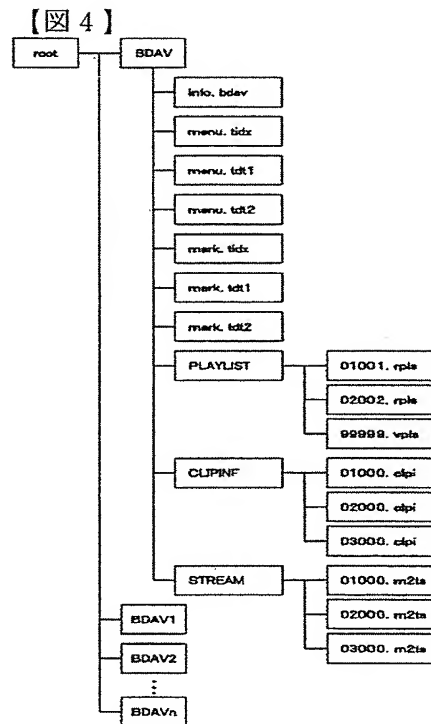
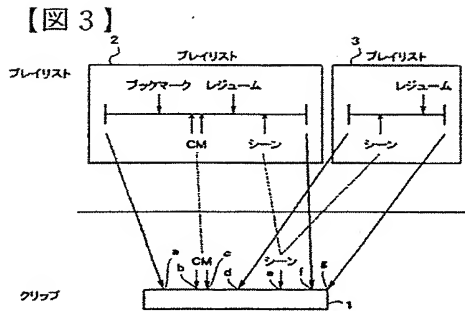
10

【図 1】



【図 2】





【図 5】

シンタックス	データ長 (ビット)	Mnemonics
info.bdav{		
type_indicator	8*4	bitbf
version_number	8*4	bitbf
TableOfPlayLists_start_address	32	unimabf
MakePrivateData_start_address	32	unimabf
reserved_for_future_use	182	bitbf
UAppInfoBDAV{		
for(i=0;i<N1;+++){		
padding_word	16	bitbf
}		
TableOfPlayLists{		
for(i=0;i<N2;+++){		
padding_word	16	bitbf
}		
MakePrivateData{		
for(i=0;i<N3;+++){		
padding_word	16	bitbf
}		
}		

【図 6】

シンタックス	データ長 (ビット)	Mnemonics
UAppInfoBDAV{		
length	32	unimabf
reserved_for_future_use	16	bitbf
BDAV_character_set	8	bitbf
reserved_for_word_align	6	bitbf
BDAV_protect_flag	1	bitbf
resume_valid_flag	1	bitbf
PIN	6*4	bitbf
resume_PlayList_file_name	8*10	bitbf
ref_to_menu_thumbnail_index	16	unimabf
BDAV_name_length	8	unimabf
BDAV_name	8*255	bitbf
}		

【図 7】

シンタックス	データ長 (ビット)	Mnemonics
TableOfPlayLists{		
length	32	unimabf
number_of_PlayLists	16	unimabf
for(i=0;i<number_of_PlayLists;+++){		
PlayList_file_name	8*10	bitbf
}		
}		

【図 8】

シンタックス	データ長 (ビット)	Mnemonics
xxxxxx.rpl/yyyy.vpls{		
type_indicator	8*4	bitbf
version_number	8*4	bitbf
Playlist_start_address	32	unimstbf
PlaylistMark_start_address	32	unimstbf
MakersPrivateData_start_address	32	unimstbf
reserved_for_future_use	160	bitbf
UAppInfoForPlaylist()		
for(i=0;i<N1;i++){		
padding_word	16	bitbf
}		
Playlist()		
for(i=0;i<N2;i++){		
padding_word	16	bitbf
}		
PlaylistMark()		
for(i=0;i<N3;i++){		
padding_word	16	bitbf
}		
MakersPrivateData()		
for(i=0;i<N4;i++){		
padding_word	16	bitbf
}		
}		

【図 9】

シンタックス	データ長 (ビット)	Mnemonics
UAppInfoForPlaylist{		
length	32	unimstbf
reserved_for_future_use	16	bitbf
Playlist_character_set	8	unimstbf
reserved_for_word_align	4	bitbf
playback_protect_flag	1	bitbf
write_protect_flag	1	bitbf
is_played_flag	1	bitbf
is_edited_flag	1	bitbf
time_zone	8	bitbf
reserved_for_word_align	8	bitbf
record_time_and_date	4*14	bitbf
Playlist_duration	4*8	bitbf
maker_ID	18	unimstbf
maker_model_code	16	unimstbf
channel_number	16	unimstbf
reserved_for_word_align	8	bitbf
channel_name_length	8	unimstbf
channel_name	8*20	bitbf
Playlist_name_length	8	unimstbf
Playlist_name	8*255	bitbf
Playlist_detail_length	16	unimstbf
Playlist_detail	8*1200	bitbf
}		

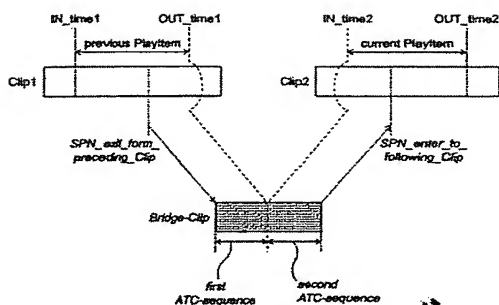
【図 10】

シンタックス	データ長 (ビット)	Mnemonics
Playlist{		
length	32	unimstbf
reserved_for_word_align	12	bitbf
PL_CPL_type	4	bitbf
number_of_PlayItems	16	unimstbf
if(<Virtual-Playlist>&&PL_CPL_type==1){		
number_of_SubPlayItems	16	unimstbf
}else{		
reserved_for_word_align	16	bitbf
}		
for(Playlist_id=0;		
Playlist_id<number_of_PlayItems;		
Playlist_id++){		
PlayItem()		
}		
if(<Virtual-Playlist>&&CPL_type==1){		
for(i=0;i<number_of_SubPlayItems;i++){		
SubPlayItem()		
}		
}		
}		

【図 11】

シンタックス	データ長 (ビット)	Mnemonics
PlayItem{		
length	16	unimstbf
Clip_informant_file_name	8*5	bitbf
Clip_codec_identifier	8*4	bitbf
reserved_for_future_use	6	bitbf
connection_condition	2	bitbf
if(CPL_type==1){		
ref_to_STC_id	8	unimstbf
}else{		
reserved_for_word_align	8	bitbf
}		
IN_time	32	unimstbf
OUT_time	32	unimstbf
if(<Virtual-Playlist>	3.5	
connection_condition==3){		
BridgeSequenceInfo()		
}		
}		

【図 12】



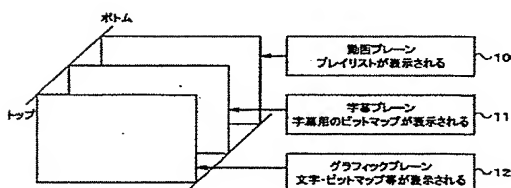
【図 13】

シンタックス	データ長 (ビット)	Mnemonics
PlaylistMark{		
length	32	uintmbf
number_of_PlayList_marks	16	uintmbf
for(i=0; i<number_of_PlayList_marks; i++){		
mark_invalid_flag	1	uintmbf
mark_type	7	uintmbf
mark_name_length	8	uintmbf
maker_ID	16	uintmbf
ref_to_PlayItem_id	16	uintmbf
mark_time_stamp	32	uintmbf
entry_ES_PID	16	uintmbf
if(mark_type==0x01){mark_type==0x02}{		
ref_to_menu_thumbnail_index	16	uintmbf
}else{		
ref_to_menu_thumbnail_index	16	uintmbf
}		
duration	32	uintmbf
makers_information	32	bitbf
mark_name	8*24	bitbf
}		

【図 14】

シンタックス	データ長 (ビット)	Mnemonics
zzzzz.cpl{		
type_indicator	8*4	bitbf
version_number	8*4	bitbf
SequenceInfo_start_address	32	uintmbf
ProgramInfo_start_address	32	uintmbf
CPI_start_address	32	uintmbf
ClipMark_start_address	32	uintmbf
MakersPrivateData_start_address	32	uintmbf
reserved_for_future_use	96	bitbf
ClipInfo{		
for(i=0; i<N1; i++){		
padding_word	16	bitbf
}		
SequenceInfo{		
for(i=0; i<N2; i++){		
padding_word	16	bitbf
}		
ProgramInfo{		
for(i=0; i<N3; i++){		
padding_word	16	bitbf
}		
CPI{		
for(i=0; i<N4; i++){		
padding_word	16	bitbf
}		
ClipMark{		
for(i=0; i<N5; i++){		
padding_word	16	bitbf
}		
MakersPrivateData{		
for(i=0; i<N6; i++){		
padding_word	16	bitbf
}		
}		

【図 15】

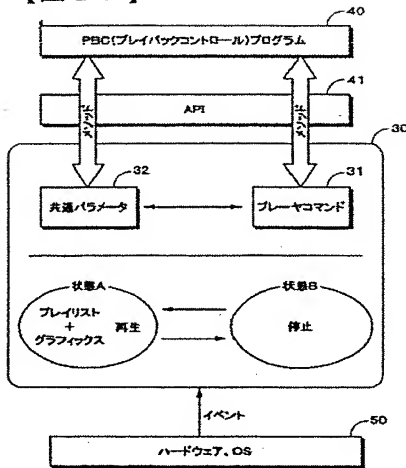


【図 16】

項目	規定内容
動画プレーン	1920x1080x16ビット, YCbCr(4:2:2)
字幕プレーン	1920x1080x8ビット, 8ビットカラーマップアドレス(パレット)+256段階の7x7アレンディング
グラフィックスプレーン	1920x1080x24ビット, YCbCr(4:4:4)またはRGB(4:4:4)各8ビット+256段階の7x7アレンディング



【図 22】



【図 23】

イベント名	説明
TimeFirst	お客がタイムファーストになった お客がタイムファースト指定の番になった
PlayStopped	再生が停止した
PlayStill	再生が一時停止した
StillReleased	一時停止が解除された
PlayPaused	コースにより一時停止された
PauseReleased	ポーズが解除された
PlayStarted	再生が開始された
PlayRepeated	繰り返し再生時の先頭を抽出した
SPDisplayStatusChanged	SPストリームの表示・非表示状態が変わった
SelectedAudioChanged	再生するAudioストリームが変わった
VideoStopped	再生するVideoストリームが変わった
ScenarioStarted	シナリオの先頭を抽出
ScenarioEnded	シナリオの終わりを抽出
PlaylistStarted	プレイリストの先頭を抽出
PlaylistEnded	プレイリストの終わりを抽出
PlaylistItemStarted	プレイリストの先頭を抽出
PlaylistItemEnded	プレイリストの終わりを抽出
MarkEncountered	再生中にマークを抽出 グラフィックス画面を表示する際に利用 共通パラメータに、種別と番号が格納される
ButtonPressed	画面上のボタンが押された 共通パラメータに、押されたもののIDが格納される
ValidPeriodStarted	有効期間が開始した リンクの選択有効期間を設定する際に利用
ValidPeriodEnded	有効期間が終了した リンクの強制実行の時に利用
KeyPressed	リモコンのキーが押された どのキーが押されたかは、イベント内でのswitchなどで識別する

【図 24】

オブジェクト	メソッド	備考
b4p		
プレーヤー動作		
	playScenario(scenarioNumber, [scenarioTime])	scenarioNumberで指定されるシナリオを再生する。 ScenarioNumberは、実際にはscenario構成を記述したファイル等を指すURIになる。 scenarioTimeはオプションであるが、シナリオの経過時間を指定する。
	playPlaylist(playlistNumber)	playlistNumberで指定されたPlaylistを再生する。
	playChapterMark(playlistNumber, chapterNumber)	playlistNumberで決まるPlaylist中のchapterNumberで指定されるあるチャプターから再生する。
	playPlayItem(playlistNumber, playItemNumber)	playlistNumberとplayItemNumberで決まるPlayItemから再生する。 playItemNumberはPlayItemであり、0から始まる。 Playlistの先頭から再生するのであれば、playItemNumberは0になる。
	playPosition(object) position = ("prev"   "next"   "top"   "goUp"   "tail") object = {Playlist   PlayItem   Chapter}	シナリオ内移動。現在再生中の箇所から前後のPlaylist/PlayItem等に移転する。
	stop()	シナリオの再生を停止する。 標準シナリオの場合は保持されない。
	resume()	前回の再生を停止した箇所から再生を再開する。
	playSoundEffect()	選択された効果音を再生

【図 25】

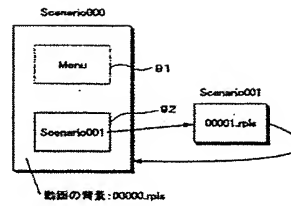
プレーヤー状態	
getMenuDescriptionLanguage()	メニュー表示の際の言語を取得
getScenarioNumber()	再生中のシナリオ番号を得る。
getPlaylistNumber()	再生中のプレイリスト番号を得る。
getChapterNumber()	再生中のチャプター番号を得る。
getPlayerSupport()	プレイヤーのバージョン、有する機能を取得
ビデオストリーム	
setVideoStreamNumber()	デコードするビデオストリームを指定する。
getVideoStreamNumber()	選択中のビデオストリーム番号を得る。
getVideoStreamStatus()	ビデオストリームの状態を得る。
getVideoStreamAttr()	ビデオストリームの属性を得る。 (符号化方式、解像度、アスペクト比、4:3時のディスプレイモード、ClosedCaption)
setAngleNumber()	アングル番号を指定する。
getAngleNumber()	選択中のアングル番号を得る。
getMaxVideoStreams()	ビデオストリームの最大数を得る。
オーディオストリーム	
getAudioStreamAvailability()	指定のオーディオストリームが含まれているかを調べる。
getAudioStreamLanguage()	指定のオーディオストリームの言語についての情報を得る。
getAudioStreamStatus()	オーディオストリームの状態(再生or非再生)を調べる。
setAudioStreamStatus()	オーディオストリームの状態(再生or非再生)をセットする。
getAudioStreamAttributes()	オーディオストリームの属性を得る。 (符号化方式、出力、Q、N)
サブビデオストリーム	
getSPStreamAvailability()	指定のSPストリームが含まれているかを調べる。
getSPStreamLanguage()	指定のSPストリームの言語を得る。
getSPDisplayStatus()	SPの表示状態を得る。
setSPDisplayStatus()	SPの表示状態を設定する。 SPの表示のON/OFF
getSpStreamAttributes()	SPの属性を得る。 4:3 or Wide用など。



【図 26】

レジスタリセット/リセット		
	clearReg()	レジスタが内蔵しているレジスタ領域(レジスタ)に対する操作。全レジスタを初期化する。
	setReg()	レジスタへの値のセット
	getReg()	レジスタから値の読み出し
タイマ		
	sleep()	指定された秒数、分、秒を待機する。
	setTimeout()	指定された秒数後に動作や処理を実行する。
	setInterval()	指定された秒数ごとに処理を実行する。
	clearTimer()	指定されたタイマIDの処理を中止する。
	pauseTimer()	指定したタイマIDのタイマを一時的に停止する。
	resumeTimer()	指定したタイマIDのタイマを一時的に再開する。
キー入力		
	getPressedKey()	入力されたキーの種類を取得
グラフィクス関連		
	loadGraphics(filePath, ID)	filePathで指定されるファイルを読み込み、IDでグラフィックに名前を付ける。ただし、表示はしない。この処理は、IDで参照される。
	showGraphics(ID)	IDで指定されるグラフィクス画像を表示する。すでにloadGraphics()でプレーストに読み込まれていることが必要。
	hideGraphics(ID)	IDで指定されるグラフィクス画像を消す。
その他		
	random(input Number num)	1からnumまでの乱数を発生する。ECMAScriptのMathオブジェクトを使用せず、独自に定義する。
	catchEvent(eventName, eventHandler)	eventNameで指定されたイベントが発生した際、eventHandlerで指定した関数を実行することを登録する。

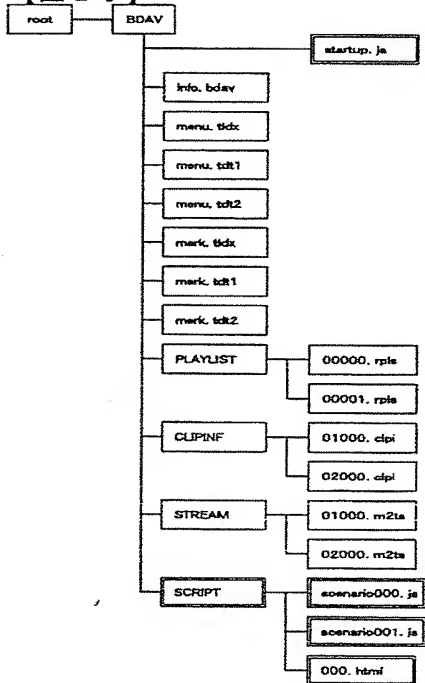
【図 27】



【図 28】

ファイル名	内容
startup.js	ファイル読み込み時に最初に実行されるスクリプトファイル
scenario000.js	シナリオ画面(シナリオ000)の構成情報を記述したスクリプトファイル
000.html	シナリオ画面のレイアウトを記述したhtmlファイル
00000.rpls	シナリオ画面表示中に背景として再生されるアニメーションファイル
scenario001.js	シナリオ001の構成情報を記述したスクリプトファイル
00001.rpls	シナリオ001で再生されるアニメーションの情報が記述されたアニメーションファイル

【図 29】



【図 30】

```

startup.js

function makeArray(n){
    this.length=n;
    for(i=0;i<n;i++){
        this[i]=null;
    }
}

//シナリオの数と名称の定義
var scenario=new makeArray(2);
scenario[0]="scenario000";
scenario[1]="scenario001";

//最初のシナリオ実行
bdp.playScenario("scenario000");
  
```

【図 3 1】

```

scenario000.js

function UOPControl() {
    var keyID=getPressedKey();
    switch(keyID){
        case menu:
            bdp.playScenario("scenario000.js");//最初に戻る。
            break;
        default:
            break;
    }
}

function playListEnded() {
    bdp.playScenario("scenario000.js");//最後まで再生し終わったら、また繰り返す。
    return 0;
}

function MarkEncountered() { //グラフィックスを読み込むタイミングを示したマーク
    bdp.loadGraphics("000.html","id1");
    return 0;
}

function ValidPeriodStarted() { //グラフィックスを表示するタイミングを示すマーク
    if("id1") {
        bdp.showGraphics("id1");
    }
    return 0;
}

function ValidPeriodEnded() { //グラフィックス表示を停止するタイミングを示すマーク
    if("id1") {
        bdp.hideGraphics("id1");
    }
    return 0;
}

bdp.catchEvent("onKeyPressed","UOPControl");
bdp.catchEvent("onPlayListEnded","PlayListEnded");
bdp.catchEvent("onMarkEncountered","MarkEncountered");
bdp.catchEvent("onValidPeriodStarted","ValidPeriodStarted");
bdp.catchEvent("onValidPeriodEnded","ValidPeriodEnded");

bdp.playPlayList("00000.rpl");

```

【図 3 2】

```

000.html

<html>
<head>
<style type="text/css">
<[CDATA[
img{menu [position: absolute; top: 200px; left: 800px; width: 200px; height: 50px;
img{scenario001 [position: absolute; top: 700px; left: 700px; width: 400px; height: 100px;
]}
</style>
<script type="text/javascript">
function onMoverHandler() {
    switch() {
        case scenario001:
            fsrc="201.png";
            break;
        default:
            break;
    }
}

function onMouthHandler() {
    switch() {
        case scenario001:
            fsrc="202.png";
            break;
        default:
            break;
    }
}

function onMouthClickHandler() {
    switch() {
        case scenario001:
            fsrc="202.png";
            bdp.playScenario("scenario001.js");
            break;
        default:
            break;
    }
}
</script>
</head>
<body>


</body>
</html>

```

【図 3 3】

```

scenario001.js

function UOPControl() {
    var keyID=getPressedKey();
    switch(keyID){
        case menu://シナリオ001再生中にメニューキーが押された
            bdp.playScenario("scenario000.js");
            break;
        default:
            break;
    }
}

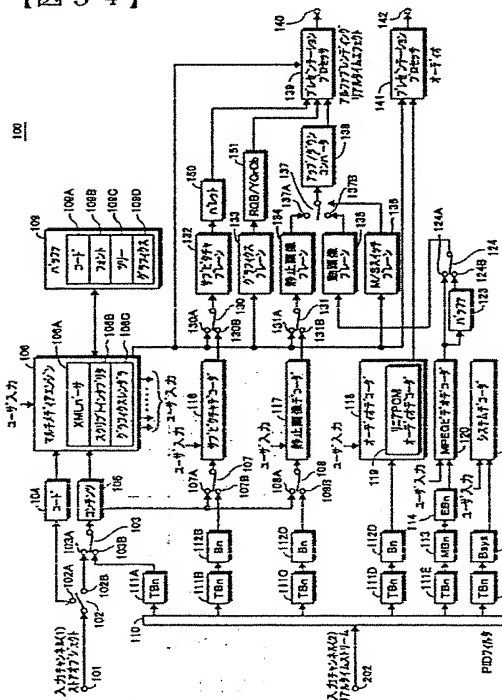
function playListEnded() { //再生が終了したので、シナリオ000(メニュー画面)に戻る。
    bdp.playScenario("scenario000.js");
    return 0;
}

bdp.catchEvent("onPlayListEnded","PlayListEnded");
bdp.catchEvent("onKeyPressed","UOPControl");

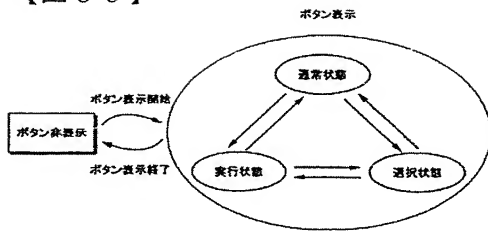
bdp.playPlayList("00001.rpl");

```

【図 3 4】



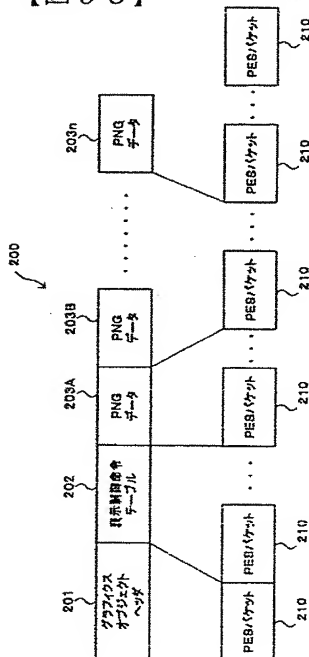
【圖 35】



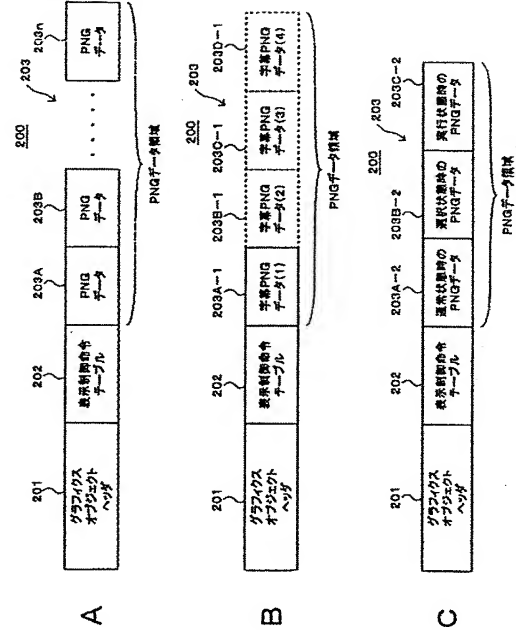
【図 3 6】

オブジェクトの種類	動画プレーンに提示される映像との関係	データ構造	表示先のプレーン <sup>*)</sup>
字幕	同期型	グラフィクス オブジェクト(GOB)	字幕プレーン
同期型グラフィクス			グラフィクスプレーン
非同同期型グラフィクス	非同同期型	任意の形式	

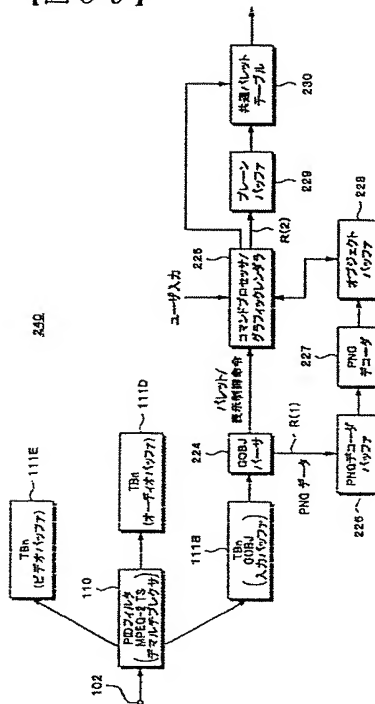
【図 38】

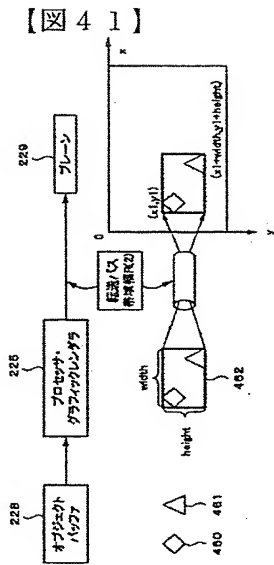
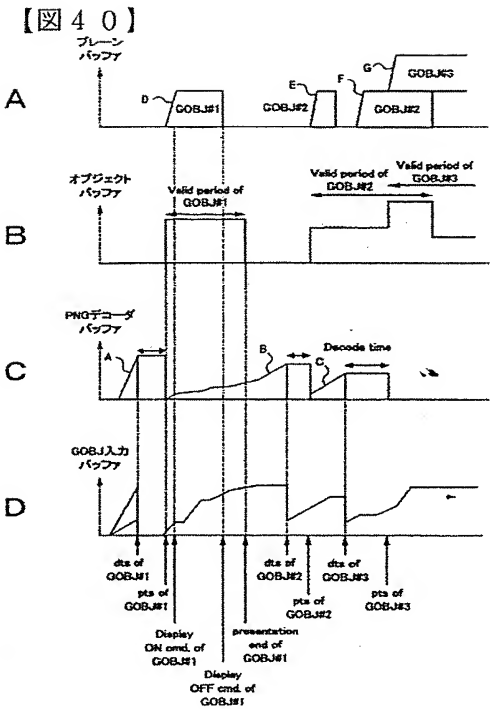


【図 37】



【圖 39】





【図 4 2】

シンタクス	データ長 (ビット)	Mnemonics
GraphicsObject{		
GraphicsObjectHeader{		
length	8	unimbsf
reserved	7	
presentation_end_time_stamp	33	unimbsf
number_of_PNG_images	8	unimbsf
number_of_DispCmds	8	unimbsf
GlobalPaletteTable{		
for(i=0; i<Number_of_PNGs; i++){		
start_address_of_PNG_image[i]	32	unimbsf
PNG_file_name[i]	6*32	bitbf
}		
for(i=0; i<Number_of_DispCmds; i++){		
start_address_of_PNG_Cmds[i]	32	unimbsf
}		
}		
for(i=0; i<N1; i++){		
padding_word	32	bitbf
}		
GOBCommandTable{		
for(i=0; i<Number_of_DispCmds; i++){		
DispCmds[i]		
}		
}		
for(i=0; i<N2; i++){		
padding_word	32	bitbf
}		
PNGImageRegion{		
for(i=0; i<Number_of_PNGs; i++){		
PNG_image[i]		
}		
}		
}		

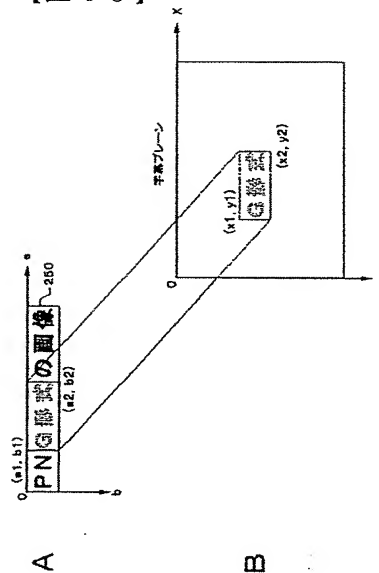
【図 4 3】

シンタクス	データ長 (ビット)	Mnemonics
GlobalPaletteTable{		
reserved_for_future_use	8	bitbf
number_of_palette_entries	8	unimbsf
for(i=0; i<number_of_palette_entries; i++){		
palette_index_number	8	unimbsf
red_value	8	unimbsf
green_value	8	unimbsf
blue_value	8	unimbsf
alpha	8	unimbsf
}		
}		

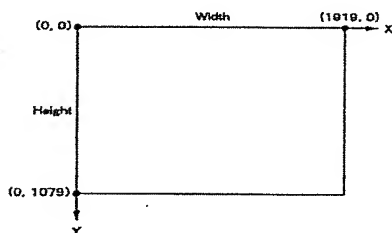
【図 4 4】

表示制御命令	内容
(1) <code>execution_time(start_time)</code>	この命令の後にあり、次の <code>execution_time(start_time)</code> の前までの命令を <code>start_time</code> で指定した時刻から実行することを指示する命令。 <code>start_time</code> の単位は、グラフィックスAPIの <code>time</code> (presentation time stamp)とする。 <code>start_time</code> の単位は <code>pts</code> と同一。
(2) <code>fade_in(fade_in_time)</code>	グラフィックスオブジェクトの表示を開始する。 現在のハレットレームの不透明度の値を徐々に小さくしていき、 <code>fade_in_time</code> 秒(2000倍)に設定する。 <code>fade_in_time</code> が0の時は、ハレットレームの色・不透明度で即時に表示される。
(3) <code>fade_out(fade_out_time)</code>	グラフィックスオブジェクトの表示を停止する命令。 不透明度の初期値 <code>0</code> から徐々に上げていき、 <code>fade_out_time</code> 秒経過後にハレットレームの値に設定する。 <code>fade_out_time</code> が0の時は、即時に表示停止する。
(4) <code>change_palette(index, newr, newg, newb, newAlpha)</code>	ハレット番号 <code>index</code> の色を <code>(newr, newg, newb)</code> に、不透明度を <code>newAlpha</code> に変更する。
(5) <code>set_display_box(x1,y1,x2,y2)</code>	プレートの <code>(x1,y1,x2,y2)</code> で定まる矩形の領域にグラフィックスオブジェクトを配置する。
(6) <code>set_clipping_box(a1,b1,a2,b2)</code>	グラフィックスオブジェクト内の <code>(a1,b1,a2,b2)</code> で定まる矩形の領域をプレート上に表示する。
(7) <code>play_sound(sound_id)</code>	<code>sound_id</code> で指定された音声データを再生する。
(8) <code>set_sound(PNG_image_id,sound_id)</code>	PNGデータが指定された音声を再生する。 PNGデータに指定された音声がPNGデータが表示される際に、 <code>sound_id</code> で指定された音声データが再生されるよう指示する命令。 PNG image IDは、シヤブス中のPNG image IDにおける(ムーブメント)と同じである。

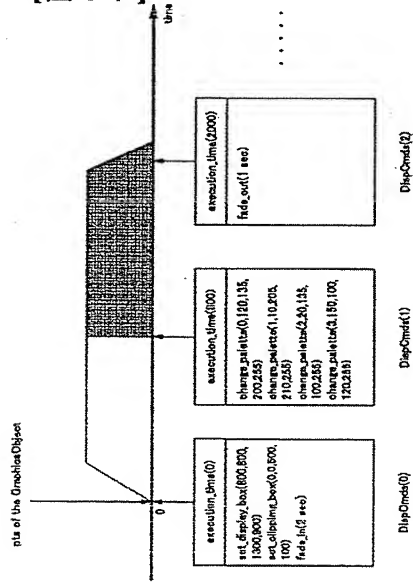
【図 45】



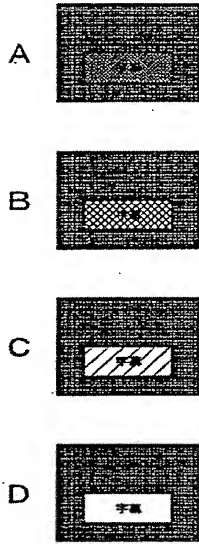
【図 46】



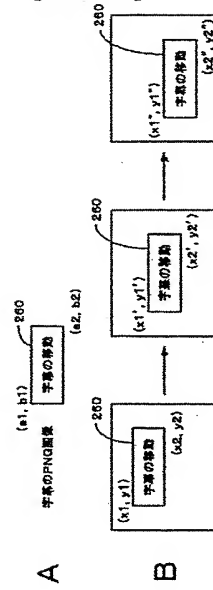
【图 4 7】



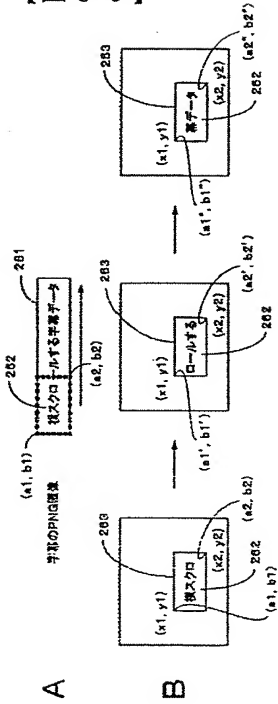
【図 48】



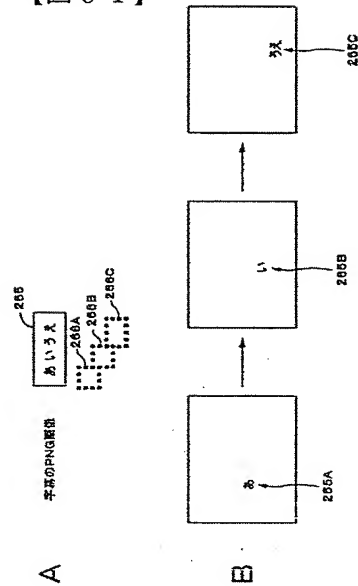
【図 49】



【図 50】



【図 51】



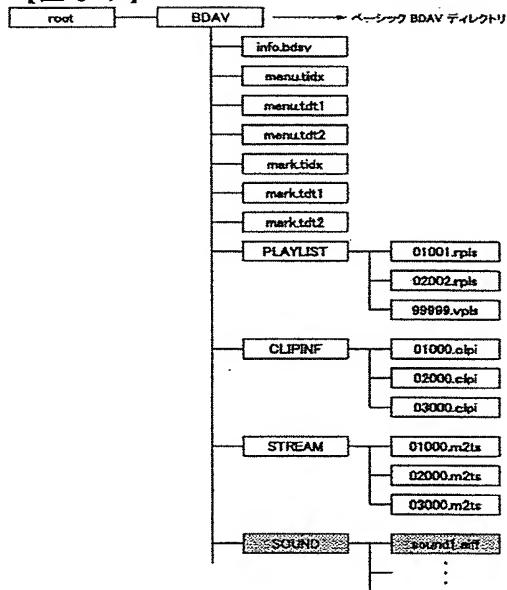




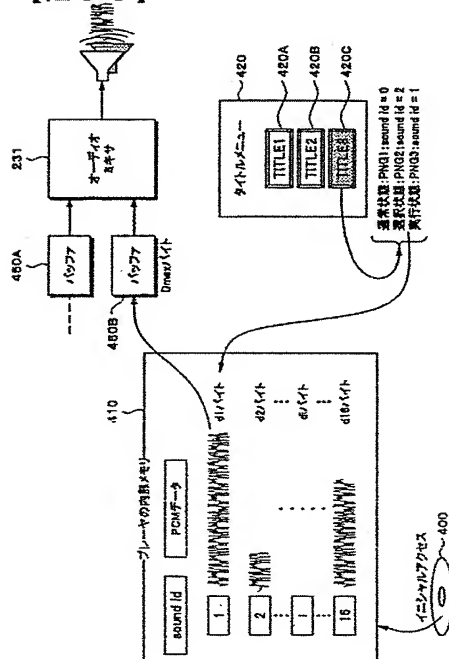
【圖 5 6】

シNTAX	データ型 (ビット)	Mnemonic
GraphicsObject{}		
GraphicsObjectHeader{}		
length	8	unimbsf
reserved	7	
presentation_end_time_stamp	33	unimbsf
number_of_DispCmds	8	unimbsf
number_of_PNG_images	8	unimbsf
number_of_sound_data (GlobalPaletteTable)	8	unimbsf
for(i=0; i<number_of_DispCmds; i++) start_address_of_DispCmds_image(i)	32	unimbsf
}		
for(i=0; i<number_of_PNG_images; i++) start_address_of_PNG_image(i) PNG_file_name(i)	32 8*32	unimbsf bsbf
}		
for(i=0; i<number_of_sound_data; i++) start_address_of_sound_data(i)	32	unimbsf
}		
for(i=0; i<N1; i++) padding_word	32	bsbf
}		
GOBCommandTable{ for(i=0; i<number_of_DispCmds; i++) DispCmds(i)		
}		
for(i=0; i<N2; i++) padding_word	32	bsbf
}		
PNGImageRegion{ for(i=0; i<number_of_PNGs; i++) PNG_image(i)		
}		
for(i=0; i<N3; i++) padding_word	32	bsbf
}		
SoundDataRegion{ for(i=0; i<number_of_sound_data; i++) sound_data(i)		
}		
}		

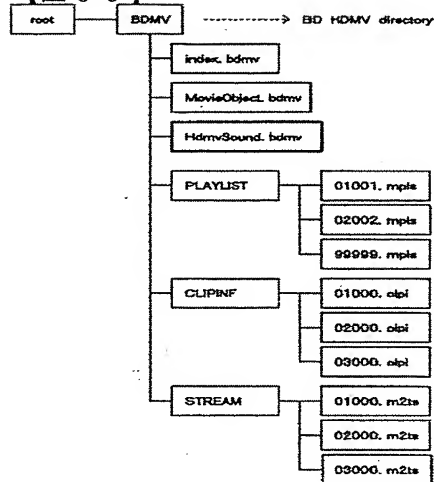
【図 5 7】



【図 5.8】



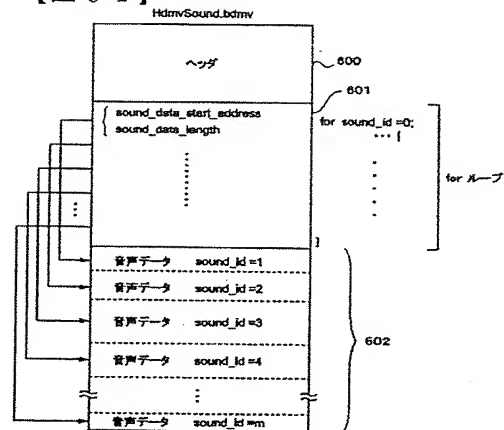
【圖 59】



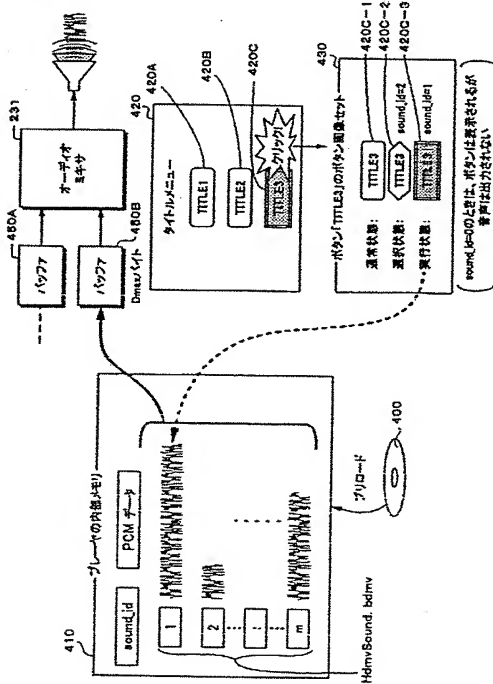
【図 60】

Syntax	No. of bits	Mnemonic
HdmvSound.bdmv		
length	32	uint8bf
if (length != 0) {		
data_block_start_address	32	uint8bf
reserved_for_word_align	24	bitbf
number_of_sound_entries	8	uint8bf
for (sound_id = 0;		
sound_id < number_of_sound_entries;		
sound_id++) {		
attributes() {		
channel_assignment		
sampling_frequency		
bits_per_sample		
}		
sound_data_start_address	32	uint8bf
sound_data_length	32	uint8bf
}		
for (i=0; i<L1; i++) {		
padding_word	16	bitbf
padding_word	16	bitbf
}		
data_block	32 + 8 * (length - data_block_start_address)	
}		

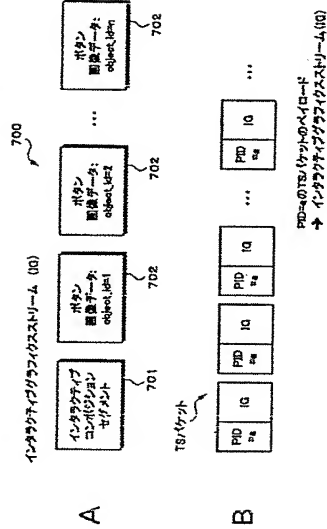
【図 61】



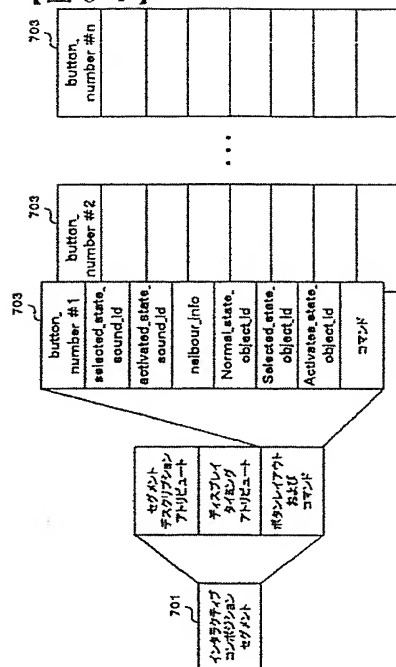
【図 62】



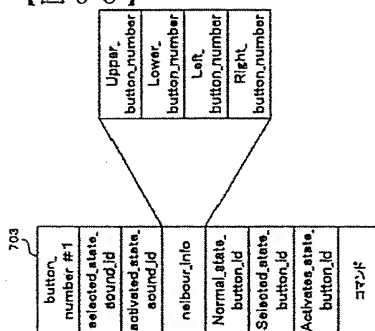
【図 63】



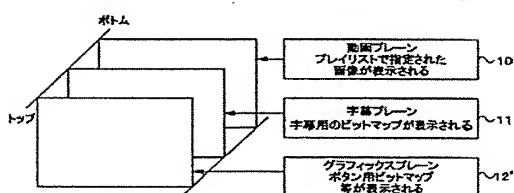
【图 6 4】



【图 65】



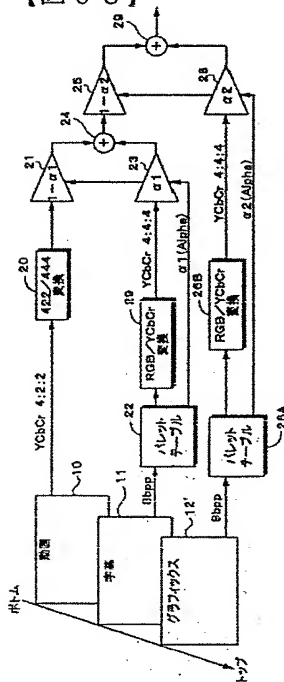
【圖 6 6】



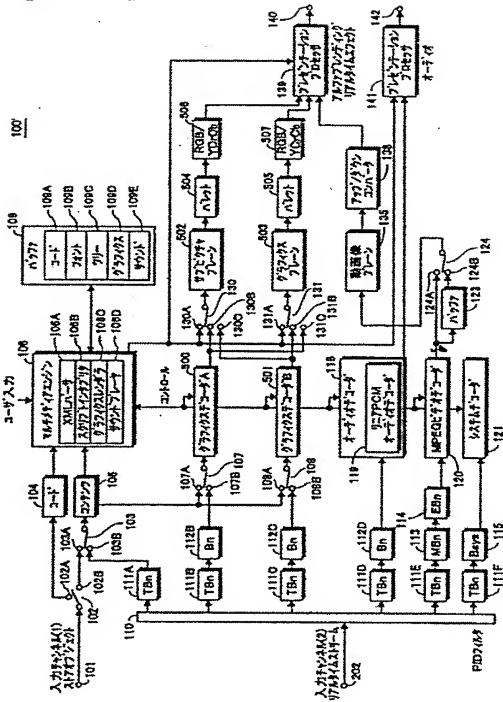
【圖 6 7】

項目	規定内容
動画プレーン	1920x1080x18ビット、YC80(4:2:2)各8ビット
字幕プレーン	1920x1080x8ビット、16ビットデータマッピング(ハイレート)+256段階のアルファブレンドイング
グラフィックスプレーン	1920x1080x8ビット、8ビットカラーマッピング(ハイレート)+256段階のアルファブレンドイング

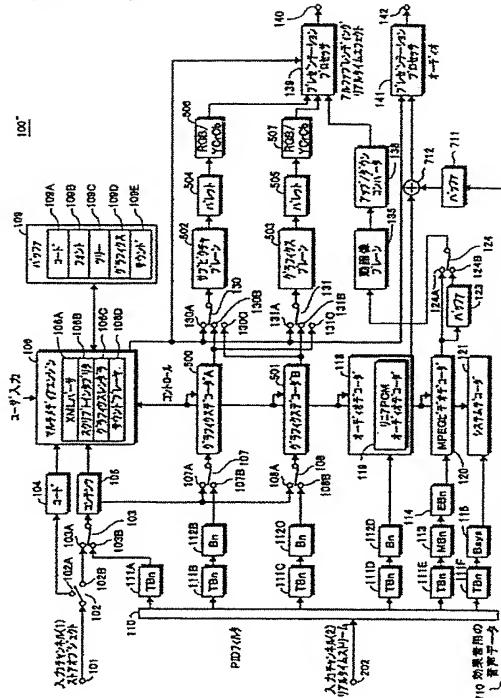
【图 68】



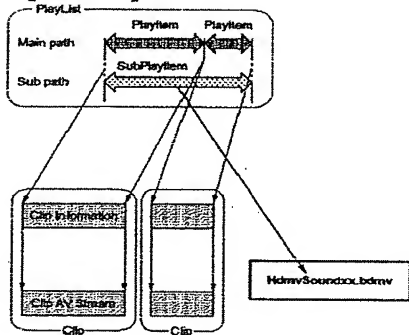
【図 69】



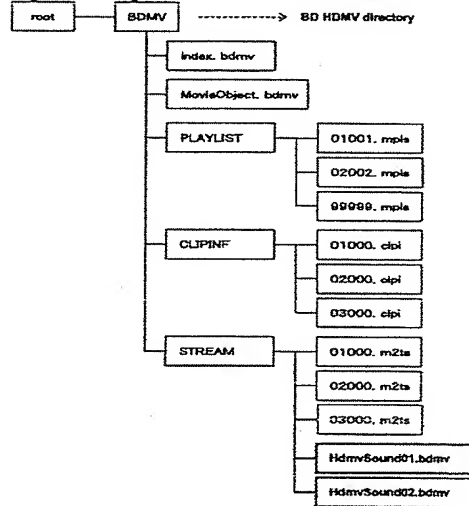
【図 70】



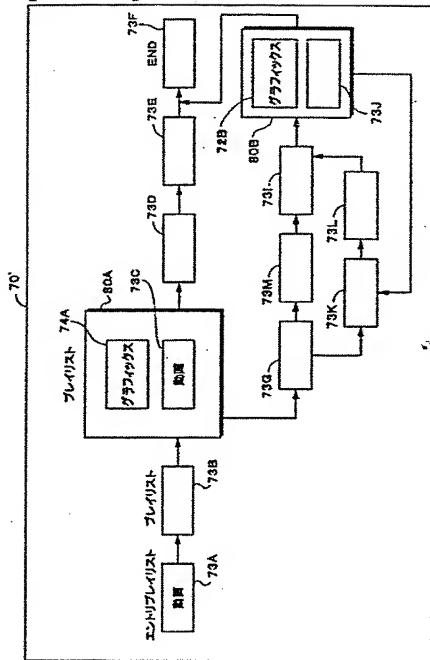
【図 71】



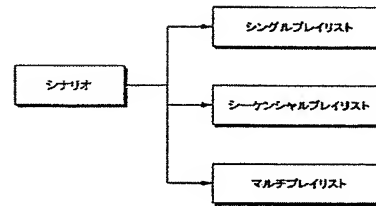
【図 72】



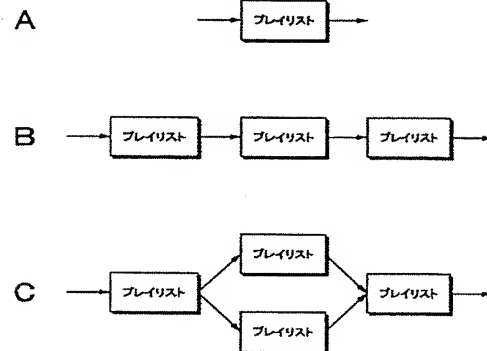
**【图 7 3】**



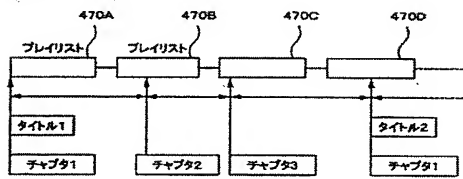
【図 7 4】



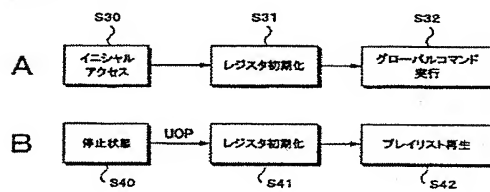
【図 7 5】



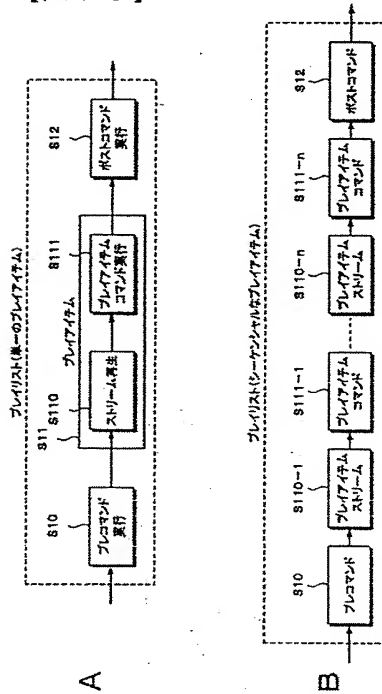
【图 7 6】



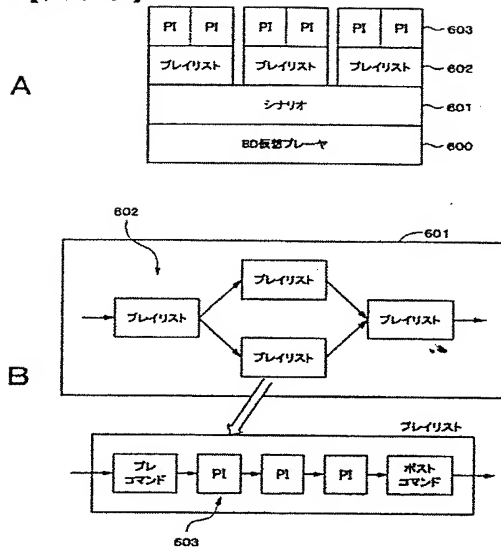
【圖 7 7】



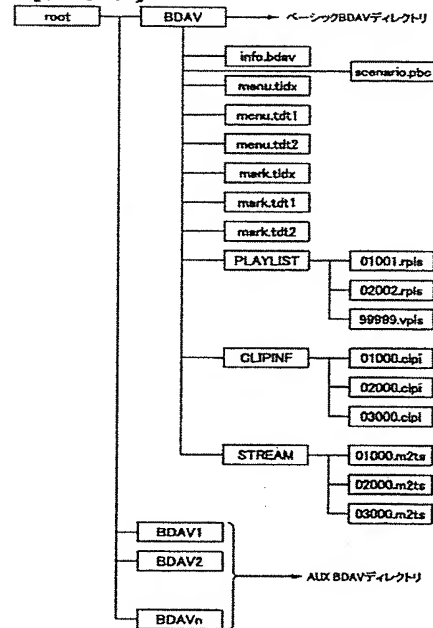
【图 78】



【図 79】



【図 80】



【図 81】

Syntax	No. of bits	Minimemo
scenario_label	8	uint8
type_indicator	8	uint8
version_number	32	uint32
TitleEntry_start_address	32	uint32
Scenario_start_address	32	uint32
reserved_for_future_use	192	uint192
GlobalCommand		
for (i=0; i<N; i++)		
padding_word	16	uint16
TitleEntry		
for (i=0; i<N; i++)		
padding_word	16	uint16
Scenario		
for (i=0; i<N; i++)		
padding_word	16	uint16

【図 82】

Syntax	No. of bits	Minimemo
GlobalCommand	32	uint32
length	16	uint16
reserved_for_word_align	16	uint16
number_of_commands	32	uint32
for (command_number=0; command_number<number_of_commands; command_number++)		
command		

【図 8 3】

Syntax	Comments	Number of bits	Remarks
length		32	default
reserved for word sign		18	default
number of PlayList		8*10	default
for (uint this number is the number of PlayList number)		8	default
Entry PlayList file name		8*155	default
This menu character set			
This menu			
//Character Emits			
number of Chapters			
for (uint chapter number is the number of Chapters number)			
This number		18	default
chapter entry PlayList file name		8*10	default
chapter entry PlayList file name		18	default
chapter entry PlayList file name		32	default
//Entry PL for the Title Menu			
Title Menu (all PL)			
PL		32	
This menu entry PlayList file name		8*10	default
This menu entry PlayList file name		18	default
This menu entry PlayList file name		32	default
UDF mask table // For Volume			

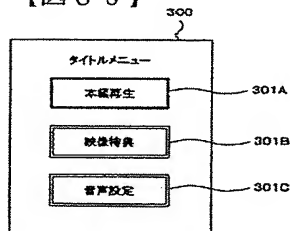
【図 8 4】

Syntax	Comments	Number of bits	Remarks
length		32	
reserved for word sign		18	
number of PlayList		18	
for (uint number of PlayList)			
PL UDF mask table // For PlayList		32	
number of PlayList		32	
number of PlayList		32	
number of PlayList		32	
number of PlayList		32	
number of PlayList		32	
for (uint number of PlayList)			
PL UDF mask table // For PlayList		32	
PL UDF mask table // For PlayList		32	
number of PlayList		18	
number of PlayList			
for (uint number of PlayList)			
PL UDF mask table // For PlayList		32	

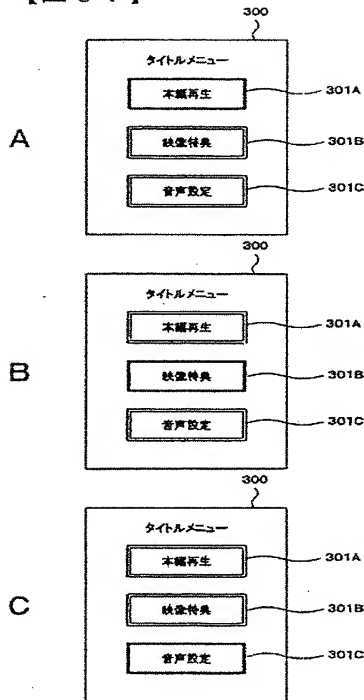
【図 8 5】

メソッド	備考
再生開始位置指定	
LinkPlayList(playListNumber)	playListNumberで指定されたPlayListの再生を開始する。
LinkPlayItem(playListNumber, playItemNumber)	指定のPlayListの指定のPlayItemから再生を開始する。 playItemNumberはPlayItemIdであり、0から始まる。 PlayListの先頭から再生するのであれば、playItemNumberは0になる。
Link(position(object) position="prev"   "next"   "top"   "parent"   "tail") object(PlayList   PlayItem   Chapter)	シナリオ内移動。 現在再生中の箇所から前後のPlayList/PlayItem, Chapterに移動する。
Exit	シナリオの再生を停止する。 結果レジスタの値は保持されない。
RSM	前回再生を停止した場所から再生を再開する。 保存されているResume情報を呼び出し、レジスタにセットして再生を開始する。
Player状態の取得	
getMenuDescriptionLanguage()	メニュー表示の際の言語を取得
getScenarioNumber()	再生中のシナリオ番号を得る。
getPlayListNumber()	再生中のプレイリスト番号を得る。
getChapterNumber()	再生中のチャプター番号を得る。
getPlayerSupport()	プレイヤーのバージョン、有する機能を取得。
Videoストリームに関するもの	
getVideoStreamAvailability()	指定のビデオストリームが含まれているかを得る。
setVideoStreamNumber()	デコードするビデオストリームを指定する。
getVideoStreamNumber()	選択中のビデオストリームの番号を得る。
getVideoStreamAttributes()	ビデオストリームの属性を得る。 (符号化方式、解像度、アスペクト比、4:3時のディスプレイモード、ClosedCaption)
setAngleNumber()	アングル番号を指定する。
getAngleNumber()	選択中のアングル番号を得る。
getMaxVideoStreams()	選択可能なビデオストリームの数を得る。
	getVideoStreamAvailability()があれば十分か？

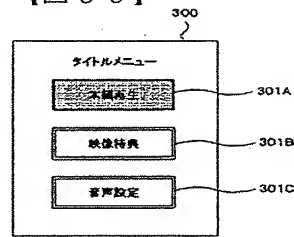
【図 8 6】



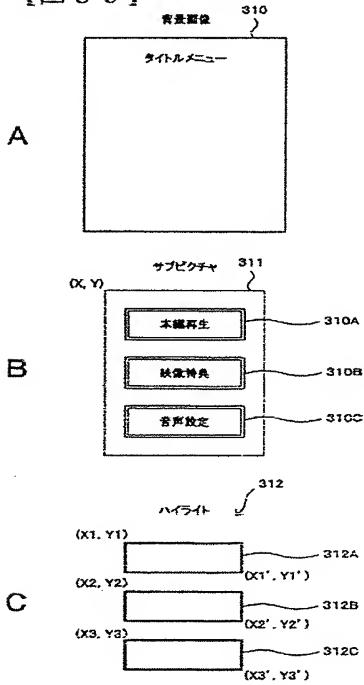
【図87】



【図88】



【図89】



【図90】

サブピクチャ	1ピクセルあたり2ビットの情報で表された一枚の 画素データ 色(A0, B0, C0, D0) 表示を開始する座標(X, Y)
ハイライト	選択状態の色(A1, B1, C1, D1) 実行状態の色(A2, B2, C2, D2) 色を変化させる領域の座標の組(=ボタンの数=3) (X1, Y1, X1', Y1') (X2, Y2, X2', Y2') (X3, Y3, X3', Y3')



## フロントページの続き

(51)Int.Cl.<sup>7</sup>

F I

テーマコード (参考)

G 1 1 B 27/34

G 1 1 B 27/34

P

H 0 4 N 5/85

H 0 4 N 5/85

Z

H 0 4 N 5/93

H 0 4 N 5/93

G

F ターム (参考) 5C053 FA20 FA23 FA29 GB01 GB05 GB11 GB12 GB37  
 5D044 AB05 AB07 AB08 BC03 CC06 DE14 DE18 DE22 DE33 DE38  
 DE49 DE52 DE57 FG10 FG18 FG23 GK12  
 5D077 AA23 BA11 BA18 CB07 CB14 DC15 EA12 HA07 HC05 HC12  
 HC17 HC25  
 5D110 AA15 AA27 BB01 CA05 CA06 CA16 CD06 CD23 CH02 CH03  
 DA01 DA04 DA12 DA19 DB03 DB08 DC05 DE01 EA07 FA08